

mandelbrot-web

Claude Heiland-Allen

2015–2019

Contents

1	c/.gitignore	2
2	c/main.c	2
3	c/Makefile	3
4	c/mandelbrot-numerics.c	4
5	COPYING	7
6	c/shell.html	19
7	hs/bin/mandelbrot-web.hs	20
8	hs/client/mandelbrot.hs	28
9	js/m-render.html	39
10	js/m-render.js	39
11	mandelbrot-web.cabal	44
12	README.md	45
13	Setup.hs	45

1 c/.gitignore

```
index.html
index.js
index.wasm
mandelbrot-numerics.js
5 mandelbrot-numerics.wasm
```

2 c/main.c

```
#include <emscripten.h>
#include <stdlib.h>
#include <string.h>

5 EM_JS(char *, input, (void), {
    var str = decodeURI(window.location.hash.substring(1));
    var bytes = lengthBytesUTF8(str) + 1;
    var ptr = _malloc(bytes);
    stringToUTF8(str, ptr, bytes);
10 return ptr;
})

EM_JS(void, output, (const char* str), {
    document.getElementById("mandelbrot-numerics").innerHTML = UTF8ToString(str);
15 })

void done(char *out, int out_size, void *arg)
{
```

```

    output(out ? out : "failed");
20    (void) out_size;
    (void) arg;
}

int main(int argc, char **argv)
25 {
    static worker_handle numerics = 0;
    if (numerics)
    {
        emscripten_destroy_worker(numerics);
30        numerics = 0;
        output("...");
    }
    numerics = emscripten_create_worker("mandelbrot-numeric.js");
    char *in = input();
35    int in_size = strlen(in) + 1;
    for (int i = 0; i < in_size; ++i) if (in[i] == ' ') in[i] = 0;
    emscripten_call_worker(numerics, "mandelbrot-numeric", in, in_size, done, 0);
    free(in);
    return 0;
40    (void) argc;
    (void) argv;
}

```

3 c/Makefile

```

## mandelbrot-web -- web interface related to the Mandelbrot set
## Copyright (C) 2015-2018 Claude Heiland-Allen
## License GPL3+ http://www.gnu.org/licenses/gpl.html

5  prefix ?= $(HOME)/opt
CC = emcc

PKGCONFIG := PKG_CONFIG_PATH="$(prefix)/lib/pkgconfig" pkg-config
COMPILE := $(CC) -std=c99 -Wall -Wextra -pedantic -fPIC -O3 -pipe -MMD '$(
    ↪ PKGCONFIG) --cflags mandelbrot-numeric'
10  LIBS    := '$(PKGCONFIG) --libs mandelbrot-numeric' -lm -lmpfr -lgmp -lm

all: index.html mandelbrot-numeric.js

clean:
15    rm -f index.html index.js index.wasm mandelbrot-numeric.js mandelbrot-
    ↪ numerics.wasm

index.html: main.c shell.html Makefile
    $(COMPILE) -s EXPORTED_FUNCTIONS="['_main']" -o index.html --shell-file ↪
    ↪ shell.html main.c

20  mandelbrot-numeric.js: mandelbrot-numeric.c Makefile
    $(COMPILE) -s EXPORTED_FUNCTIONS="['_mandelbrot_numeric']" -s ↪
    ↪ BUILD_AS_WORKER=1 -o mandelbrot-numeric.js mandelbrot-numeric.c ↪
    ↪ $(LIBS)

.SUFFIXES:
.PHONY: all clean
25

```

```
-include $(DEPENDS)
```

4 c/mandelbrot-numeric.c

```
#define _POSIX_C_SOURCE 200809L
```

```
#include <emscripten.h>
```

```
5 #include <mandelbrot-numeric.h>
```

```
#include <errno.h>
```

```
#include <stdbool.h>
```

```
#include <stdlib.h>
```

```
10 #include <string.h>
```

```
#include <mpfr.h>
```

```
static inline bool arg_precision(const char *arg, bool *native, int *bits) {
15     if (0 == strcmp("double", arg)) {
        *native = true;
        *bits = 53;
        return true;
    } else {
20         char *check = 0;
        errno = 0;
        long int li = strtol(arg, &check, 10);
        bool valid = ! errno && arg != check && ! *check;
        int i = li;
25         if (valid && i > 1) {
            *native = false;
            *bits = i;
            return true;
        }
30     }
    return false;
}
```

```
static inline bool arg_double(const char *arg, double *x) {
35     char *check = 0;
    errno = 0;
    double d = strtod(arg, &check);
    if (! errno && arg != check && ! *check) {
        *x = d;
40     }
    return true;
}
return false;
}
```

```
45 static inline bool arg_int(const char *arg, int *x) {
    char *check = 0;
    errno = 0;
    long int li = strtol(arg, &check, 10);
    if (! errno && arg != check && ! *check) {
50         *x = li;
        return true;
    }
    return false;
}
```

```

}
55 static inline bool arg_rational(const char *arg, mpq_t x) {
    int ok = mpq_set_str(x, arg, 10);
    mpq_canonicalize(x);
    return ok == 0;
60 }

static inline bool arg_mpfr(const char *arg, mpfr_t x) {
    return 0 == mpfr_set_str(x, arg, 10, MPFR_RNDN);
}
65 static inline bool arg_mpc(const char *re, const char *im, mpc_t x) {
    int ok
        = mpfr_set_str(mpc_realref(x), re, 10, MPFR_RNDN)
        + mpfr_set_str(mpc_imagref(x), im, 10, MPFR_RNDN);
70 return ok == 0;
}

static const double twopi = 6.283185307179586;

75 #define BOOK "https://mathr.co.uk/mandelbrot/book-draft/"

extern void mandelbrot_numerics(char *in, int in_size)
{
#define FAIL do{char *s=strdup(usage);emscripten_worker_respond(s,strlen(s)+1);↵
    ↵ free(s);return;}while(0)
80 // split arguments
    int argc = 0;
    for (int i = 0; i < in_size; ++i)
    {
85         argc += (in[i] == 0);
    }
    if (! (argc > 0)) { emscripten_worker_respond(0,0); return; }
    char *argv[argc];
    char *arg = in;
90 int j = 0;
    for (int i = 0; i < in_size; ++i)
    {
        if (in[i] == 0)
        {
95             argv[j++] = arg;
            arg = &in[i + 1];
        }
    }
    if (j != argc) { emscripten_worker_respond(0,0); return; }
100 if (! strcmp("nucleus", argv[0])) {
    const char *usage =
        "<h1><a href='#' title='mandelbrot-numerics'>mandelbrot-numerics</a></h1>\n"
        ↵ n"
        "<h2><a href='#nucleus' title='nucleus'>nucleus</a></h2>\n"
105 "<p><a href='\" BOOK \"#nucleus' title='nucleus'>nucleus in mandelbook (↵
        ↵ draft)</a></p>\n"
        "<h3>usage</h3>\n"
        "<dl>\n"

```

```

    " <dt>precision</dt><dd>can be either 'double' or a number of bits</dd>\n↵
    " <dt>cre</dt><dd>floating point real coordinate of the complex 'c' ↵
    " <dt>cim</dt><dd>floating point imaginary coordinate of the complex 'c' ↵
110    " <dt>period</dt><dd>integer period of the target hyperbolic component</↵
    " <dt>maxsteps</dt><dd>maximum number of steps of Newton's method to ↵
    " </dl>\n"
    "<h3>examples</h3>\n"
115    "<ul>\n"
    " <li><a href='#nucleus double 0 1 10 16' title='#nucleus double 0 1 10 ↵
    " <li><a href='#nucleus 1000 -2 0 250 16' title='#nucleus 1000 -2 0 250 ↵
    " </ul>\n"
    ;
120    if (argc != 6) FAIL;
    bool native = true;
    int bits = 0;
    if (! arg_precision(argv[1], &native, &bits)) FAIL;
    if (native) {
125        double cre = 0;
        double cim = 0;
        int period = 0;
        int maxsteps = 0;
        if (! arg_double(argv[2], &cre)) FAIL;
130        if (! arg_double(argv[3], &cim)) FAIL;
        if (! arg_int(argv[4], &period)) FAIL;
        if (! arg_int(argv[5], &maxsteps)) FAIL;
        double _Complex c = 0;
        m_d_nucleus(&c, cre + I * cim, period, maxsteps);
135        const char *result =
            "<h1><a href='#' title='mandelbrot-numeric'>mandelbrot-numeric</a></h1>↵
            " <h2><a href='#nucleus' title='nucleus'>nucleus</a></h2>\n"
            "<h3>input</h3>\n<p id='input'><a href='#%s %s %s %s %s %s' title='#%s %s↵
            " <h3>output</h3>\n<textarea id='output'>%s %s</textarea>\n"
140        ;
        int out_bytes = strlen(result) + 3 * in_size + 100;
        char *out = malloc(out_bytes);
        snprintf
145            ( out, out_bytes
              , result
              , argv[0], argv[1], argv[2], argv[3], argv[4], argv[5]
              , argv[0], argv[1], argv[2], argv[3], argv[4], argv[5]
              , argv[0], argv[1], argv[2], argv[3], argv[4], argv[5]
              , creal(c), cimag(c)
150            );
        int out_size = strlen(out, out_bytes) + 1;
        emscripten_worker_respond(out, out_size);
        free(out);
        return;
155    } else {

```

```

    mpc_t c_guess;
    int period = 0;
    int maxsteps = 0;
    mpc_init2(c_guess, bits);
160   if (! arg_mpc(argv[2], argv[3], c_guess)) FAIL;
    if (! arg_int(argv[4], &period)) FAIL;
    if (! arg_int(argv[5], &maxsteps)) FAIL;
    mpc_t c_out;
    mpc_init2(c_out, bits);
165   m_r_nucleus(c_out, c_guess, period, maxsteps, 0);
    const char *result =
        "<h1><a href='#' title='mandelbrot-numerics'>mandelbrot-numerics</a></h1>\n"
        "\n"
        "<h2><a href='#nucleus' title='nucleus'>nucleus</a></h2>\n"
        "<h3>input</h3>\n<p id='input'><a href='#%s %s %s %s %s %s' title='#%s %s"
        "\n"
        "\n"
170   "<h3>output</h3>\n<textarea id='output'>%Re %Re</textarea>\n"
        ;
    int out_size = mpfr_snprintf
        ( 0, 0
          , result
175   , argv[0], argv[1], argv[2], argv[3], argv[4], argv[5]
          , argv[0], argv[1], argv[2], argv[3], argv[4], argv[5]
          , argv[0], argv[1], argv[2], argv[3], argv[4], argv[5]
          , mpc_realref(c_out), mpc_imagref(c_out)
          ) + 1;
180   char *out = malloc(out_size);
    mpfr_snprintf
        ( out, out_size
          , result
185   , argv[0], argv[1], argv[2], argv[3], argv[4], argv[5]
          , argv[0], argv[1], argv[2], argv[3], argv[4], argv[5]
          , argv[0], argv[1], argv[2], argv[3], argv[4], argv[5]
          , mpc_realref(c_out), mpc_imagref(c_out)
          );
    emscripten_worker_respond(out, out_size);
190   free(out);
    mpc_clear(c_out);
    mpc_clear(c_guess);
    return;
}
195 }

const char *usage =
    "<h1><a href='#' title='mandelbrot-numerics'>mandelbrot-numerics</a></h1>\n"
    "<h2>available commands</h2>\n"
200   "<h3><a href='#nucleus' title='nucleus'>nucleus</a></h3>\n"
        ;
    FAIL;
#undef FAIL
}

```

5 COPYING

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program—to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights. Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received. You must make sure that they, too, receive
or can get the source code. And you must show them these terms so they
know their rights.

Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software. For both users' and
authors' sake, the GPL requires that modified versions be marked as
changed, so that their problems will not be attributed erroneously to
authors of previous versions.

Some devices are designed to deny users access to install or run
modified versions of the software inside them, although the manufacturer
can do so. This is fundamentally incompatible with the aim of
protecting users' freedom to change the software. The systematic
pattern of such abuse occurs in the area of products for individuals to
use, which is precisely where it is most unacceptable. Therefore, we
have designed this version of the GPL to prohibit the practice for those
products. If such problems arise substantially in other domains, we
stand ready to extend this provision to those domains in future versions
of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

175 Conveying under any other circumstances is permitted solely under
the conditions stated below. Sublicensing is not allowed; section 10
makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

180 No covered work shall be deemed part of an effective technological
measure under any applicable law fulfilling obligations under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or
similar laws prohibiting or restricting circumvention of such
185 measures.

When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
190 the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.

4. Conveying Verbatim Copies.

195 You may convey verbatim copies of the Program's source code as you
receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice;
200 keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

205 You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

210 You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

215 a) The work must carry prominent notices stating that you modified
it, and giving a relevant date.

b) The work must carry prominent notices stating that it is
released under this License and any conditions added under section
7. This requirement modifies the requirement in section 4 to
220 "keep intact all notices".

c) You must license the entire work, as a whole, under this
License to anyone who comes into possession of a copy. This
License will therefore apply, along with any applicable section 7
225 additional terms, to the whole of the work, and all its parts,
regardless of how they are packaged. This License gives no
permission to license the work in any other way, but it does not
invalidate such permission if you have separately received it.

230 d) If the work has interactive user interfaces, each must display
Appropriate Legal Notices; however, if the Program has interactive

interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

235 A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not
240 used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

245 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License,
250 in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium
255 customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as
260 long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no
265 more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This
270 alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to
280 copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the
285 Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided

290 you inform other peers where the object code and Corresponding
Source of the work are being offered to the general public at no
charge under subsection 6d.

295 A separable portion of the object code, whose source code is excluded
from the Corresponding Source as a System Library, need not be
included in conveying the object code work.

300 A "User Product" is either (1) a "consumer product", which means any
tangible personal property which is normally used for personal, family,
or household purposes, or (2) anything designed or sold for incorporation
into a dwelling. In determining whether a product is a consumer product,
doubtful cases shall be resolved in favor of coverage. For a particular
product received by a particular user, "normally used" refers to a
typical or common use of that class of product, regardless of the status
of the particular user or of the way in which the particular user
305 actually uses, or expects or is expected to use, the product. A product
is a consumer product regardless of whether the product has substantial
commercial, industrial or non-consumer uses, unless such uses represent
the only significant mode of use of the product.

310 "Installation Information" for a User Product means any methods,
procedures, authorization keys, or other information required to install
and execute modified versions of a covered work in that User Product from
a modified version of its Corresponding Source. The information must
suffice to ensure that the continued functioning of the modified object
315 code is in no case prevented or interfered with solely because
modification has been made.

320 If you convey an object code work under this section in, or with, or
specifically for use in, a User Product, and the conveying occurs as
part of a transaction in which the right of possession and use of the
User Product is transferred to the recipient in perpetuity or for a
fixed term (regardless of how the transaction is characterized), the
Corresponding Source conveyed under this section must be accompanied
by the Installation Information. But this requirement does not apply
325 if neither you nor any third party retains the ability to install
modified object code on the User Product (for example, the work has
been installed in ROM).

330 The requirement to provide Installation Information does not include a
requirement to continue to provide support service, warranty, or updates
for a work that has been modified or installed by the recipient, or for
the User Product in which it has been modified or installed. Access to a
network may be denied when the modification itself materially and
adversely affects the operation of the network or violates the rules and
335 protocols for communication across the network.

340 Corresponding Source conveyed, and Installation Information provided,
in accord with this section must be in a format that is publicly
documented (and with an implementation available to the public in
source code form), and must require no special password or key for
unpacking, reading or copying.

7. Additional Terms.

345 "Additional permissions" are terms that supplement the terms of this

License by making exceptions from one or more of its conditions.

Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions
350 apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option
355 remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

360 Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

365 a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal
370 Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

375 d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some
380 trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for
385 any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you
390 received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms
395 of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you
400 must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions;
405 the above requirements apply either way.

8. Termination.

410 You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

415 However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means
420 prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have
425 received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

430 Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

435 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission
440 to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

445 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and
450 propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an
455 organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the

460 Corresponding Source of the work from the predecessor in interest, if
the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the
rights granted or affirmed under this License. For example, you may
465 not impose a license fee, royalty, or other charge for exercise of
rights granted under this License, and you may not initiate litigation
(including a cross-claim or counterclaim in a lawsuit) alleging that
any patent claim is infringed by making, using, selling, offering for
sale, or importing the Program or any portion of it.

470 11. Patents.

A "contributor" is a copyright holder who authorizes use under this
License of the Program or a work on which the Program is based. The
475 work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims
owned or controlled by the contributor, whether already acquired or
hereafter acquired, that would be infringed by some manner, permitted
480 by this License, of making, using, or selling its contributor version,
but do not include claims that would be infringed only as a
consequence of further modification of the contributor version. For
purposes of this definition, "control" includes the right to grant
patent sublicenses in a manner consistent with the requirements of
485 this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free
patent license under the contributor's essential patent claims, to
make, use, sell, offer for sale, import and otherwise run, modify and
490 propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express
agreement or commitment, however denominated, not to enforce a patent
(such as an express permission to practice a patent or covenant not to
495 sue for patent infringement). To "grant" such a patent license to a
party means to make such an agreement or commitment not to enforce a
patent against the party.

If you convey a covered work, knowingly relying on a patent license,
500 and the Corresponding Source of the work is not available for anyone
to copy, free of charge and under the terms of this License, through a
publicly available network server or other readily accessible means,
then you must either (1) cause the Corresponding Source to be so
available, or (2) arrange to deprive yourself of the benefit of the
505 patent license for this particular work, or (3) arrange, in a manner
consistent with the requirements of this License, to extend the patent
license to downstream recipients. "Knowingly relying" means you have
actual knowledge that, but for the patent license, your conveying the
covered work in a country, or your recipient's use of the covered work
510 in a country, would infringe one or more identifiable patents in that
country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or
arrangement, you convey, or propagate by procuring conveyance of, a
515 covered work, and grant a patent license to some of the parties
receiving the covered work authorizing them to use, propagate, modify

or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered

version or of any later version published by the Free Software
575 Foundation. If the Program does not specify a version number of the
GNU General Public License, you may choose any version ever published
by the Free Software Foundation.

If the Program specifies that a proxy can decide which future
580 versions of the GNU General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

Later license versions may give you additional or different
585 permissions. However, no additional obligations are imposed on any
author or copyright holder as a result of your choosing to follow a
later version.

15. Disclaimer of Warranty.

590 THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
595 THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

600 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
605 GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
610 SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided
615 above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

620 END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

625 If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
630 to attach them to the start of each source file to most effectively

state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```

635     <one line to give the program's name and a brief idea of what it does.>
        Copyright (C) <year> <name of author>

        This program is free software: you can redistribute it and/or modify
        it under the terms of the GNU General Public License as published by
        the Free Software Foundation, either version 3 of the License, or
640     (at your option) any later version.

        This program is distributed in the hope that it will be useful,
        but WITHOUT ANY WARRANTY; without even the implied warranty of
        MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
645     GNU General Public License for more details.

        You should have received a copy of the GNU General Public License
        along with this program. If not, see <http://www.gnu.org/licenses/>.

650     Also add information on how to contact you by electronic and paper mail.

        If the program does terminal interaction, make it output a short
        notice like this when it starts in an interactive mode:

655     <program> Copyright (C) <year> <name of author>
        This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
        This is free software, and you are welcome to redistribute it
        under certain conditions; type 'show c' for details.

660     The hypothetical commands 'show w' and 'show c' should show the appropriate
        parts of the General Public License. Of course, your program's commands
        might be different; for a GUI interface, you would use an "about box".

        You should also get your employer (if you work as a programmer) or school,
665     if any, to sign a "copyright disclaimer" for the program, if necessary.
        For more information on this, and how to apply and follow the GNU GPL, see
        <http://www.gnu.org/licenses/>.

        The GNU General Public License does not permit incorporating your program
670     into proprietary programs. If your program is a subroutine library, you
        may consider it more useful to permit linking proprietary applications with
        the library. If this is what you want to do, use the GNU Lesser General
        Public License instead of this License. But first, please read
        <http://www.gnu.org/philosophy/why-not-lgpl.html>.

```

6 c/shell.html

```

<!DOCTYPE html>
<html lang="en-us">
  <head>
    <meta charset="utf-8">
5    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>mandelbrot-numerics</title>
    <style>
      #mandelbrot-numerics { width: 80em; margin: auto; }
      textarea { width: 80em; height: 24em; margin: auto; }
10   </style>

```

```

    </head>
    <body>
      <div id="mandelbrot-numerics">...</div>
      <script type='text/javascript'>
15      var Module = {
        preRun: [],
        postRun: [],
        print: function(text) {
          if (arguments.length > 1) text = Array.prototype.slice.call(arguments) ↵
            ↵ .join(' ');
20          console.log(text);
        },
        printErr: function(text) {
          if (arguments.length > 1) text = Array.prototype.slice.call(arguments) ↵
            ↵ .join(' ');
          console.error(text);
25        },
        totalDependencies: 0,
        monitorRunDependencies: function(left) {
          this.totalDependencies = Math.max(this.totalDependencies, left);
        }
30      };
      window.onhashchange = function() { Module._main(0, 0); };
    </script>
    {{{ SCRIPT }}}
  </body>
35 </html>

```

7 hs/bin/mandelbrot-web.hs

```

{-# LANGUAGE FlexibleContexts #-}
{-# LANGUAGE OverloadedStrings #-}
import Prelude hiding (Rational)
import Network.Wai
5 import Network.HTTP.Types
import Network.Wai.Handler.Warp (run)
import Data.Bits (bit)
import Data.Char (ord)
import Data.List (nub)
10 import Data.Maybe (isJust)
import Data.Text (unpack)
import Data.Text.Encoding (decodeUtf8 ')
import Data.Text.Lazy (pack)
import Data.Text.Lazy.Encoding (encodeUtf8)
15 import Text.Parsec
  ( choice
  , try
  , getPosition
  , manyTill
20  , anyChar
  , eof
  , setPosition
  , unexpected
  , runParser
25  )
import Control.Monad.Writer
  ( execWriter

```

```

    , tell
    )
30 import Data.Maybe
    ( isNothing
    )

import Mandelbrot.Symbolics
35   ( AngledAddress(..)
    , BinaryAngle(..)
    , Block(..)
    , ExternalAngle(..)
    , InternalAddress(..)
40   , InternalAngle(..)
    , Kneading(..)
    , Rational
    , Periods(..)
    , (!)
45   , kneading
    , associated
    , rational
    , binary
    , binaryAngle
50   , compact
    , addressAngles
    , externalAngles
    , denominator
    , zero
55   , one
    , preperiod
    , period
    , safePeriods
    , periods
60   , internalAddress
    , angledAddress
    )
import Mandelbrot.Text
    ( Parser
65   , Parse(parser)
    , Plain(plain)
    , HTML(html)
    , Glyph(ArrowRight, Star)
    )
70
data Item
    = BAn BinaryAngle
    | Knd Kneading
    | AAd AngledAddress
75   | IAd InternalAddress
    | Rat Rational
    | Per Periods
    | IAn InternalAngle
    | EAn ExternalAngle
80

item :: Parser Item
item = choice
    [ t (BAn 'fmap' parser) -- starts with ., ends with )
    , t (Knd 'fmap' parser) -- starts with 0, 1, *, (, ends with )

```

```

85     , t (Rat 'fmap' parser) -- contains over
      , t (Per 'fmap' parser) -- contains p
      , t (AAd 'fmap' parser) -- possibly contains -, over, arrowright
      , t (IAd 'fmap' parser) -- possibly contains arrowright, no over
      , do
90         p <- getPosition
          s <- manyTill anyChar eof
          setPosition p
          unexpected . show . abbreviate 50 $ s
    ]
95  where
      t p = try (do r <- p ; eof ; return r)
      abbreviate n s = case map (splitAt n) (lines s) of
        [(s', [])] -> s'
        (s', _):- -> s' ++ "... "
100     _ -> ""

class Valid t where
  valid :: t -> Bool
105

instance Valid AngledAddress where
  valid (Unangled l) = True
  valid (Angled l r' a') = go l r' a'
    where
110      go p r (Unangled pp) =
        zero < r && r < one && p < pp && pp <= p * fromInteger (denominator r)
      go p r (Angled pp r1 a1) =
        zero < r && r < one && p < pp && pp <= p * fromInteger (denominator r) ✓
        ↳ &&
      go pp r1 a1
115  valid _ = False

realizable :: AngledAddress -> Bool
realizable a = valid a && all (isJust . addressAngles) (truncations a)

120 truncations :: AngledAddress -> [AngledAddress]
truncations a' = execWriter $ go (Unangled l) a'
  where
    go a0 (Unangled _) = tell [a0]
    go a0 (Angled p r (Unangled pp)) = tell [a0,a1,a2]
125    where
      a1 = appendAddress a0 r (p * fromInteger (denominator r))
      a2 = appendAddress a0 r pp
    go a0 (Angled p r a3@(Angled pp _ _)) = tell [a0,a1] >> go a2 a3
    where
130      a1 = appendAddress a0 r (p * fromInteger (denominator r))
      a2 = appendAddress a0 r pp

class Plain t => Hyper t where
  hyper :: t -> String
135  link :: t -> String -> String
  link t s = link' t (stringToHtmlString s)
  link' :: t -> String -> String
  link' t s = "<a href=?q=?" ++ q ++ "' title=" ++ q ++ "'>" ++ s ++ "</a>"
    where q = stringToHtmlString (plain t)
140

```

```

class Page t where
  page :: t -> String

instance Page AngledAddress where
145   page a0
      | realizable a0 = unlines $
        [ "<table>" ] ++
        map tr (reverse (nub (truncations a0))) ++
        [ "</table>" ]
150   | otherwise = ""
      where
        tr a = "<tr><td>" ++ hyper a ++ "</td><td><span class='m-stack'><span>" ++
          ↵ hyper (binary lo) ++ "</span><span>" ++ hyper (binary hi) ++ "</
          ↵ span></span></td></tr>"
        where
          Just (lo, hi) = addressAngles a

155 instance Page BinaryAngle where
  page b = unlines $
    [ "<h4>Binary</h4>"
    , link ' b (html b)
160   , "<h4>Rational</h4>"
    , link ' r (html r)
    , "<h4>Periods</h4>"
    , hyper (Periods (periods b))
    , "<h4>Kneading</h4>"
165   , link ' k (html k)
    , case i' of Just i -> "<h4>Internal Address</h4>\n" ++ link ' i (html i) ; ↵
      ↵ Nothing -> ""
    , case a' of Just a -> "<h4>Angled Address</h4>\n" ++ link ' a (html a) ; ↵
      ↵ Nothing -> ""
    ] ++ if null (drop 1 es) then [] else
      [ "<h4>Landings</h4>"
170   , "<table>" ] ++
      map tr es ++
      [ "</table>" ]
    where
      r = rational b
175   k = kneading r
      i' = internalAddress k
      a' = angledAddress r
      es = externalAngles b
      tr e = "<tr>" ++ (if b == e then " class='m-strong'" else "") ++ "><td>" ++
        ↵ hyper e ++ "</td><td>" ++ hyper (rational e) ++ "</td></tr>"

180 instance Page InternalAngle where
  page i = unlines $
    [ "<h4>Rational</h4>"
    , link ' i (html i)
185   , "<h4>Bulb</h4>"
    , "<table>" ] ++
    map tr [lo, hi] ++
    [ "</table>"
190   {-
    , "<h4>Spokes</h4>"
    , "<table>"
    , map tr spokes
  -}

```

```

    , "</table>"
-}
195 ]
    where
        Just (lo , hi) = addressAngles (Angled 1 i (Unangled (fromInteger (2
            ↳ denominator i))))
        tr e = "<tr><td>" ++ hyper (binary e) ++ "</td><td>" ++ hyper e ++ "</td>↳
            ↳ ></tr>"

200 instance Page Kneading where
    page k@(StarPeriodic _) = unlines $
        [ "<h4>" ++ html Star ++ "-Periodic</h4>"
        , link ' k (html k)
        , "<h4>Periods</h4>"
205 , hyper (Periods (periods k))
        ] ++ case associated k of
            Just (u, l) ->
                [ "<h4>Upper</h4>"
                , link ' u (html u)
210 , "<h4>Lower</h4>"
                , link ' l (html l)
                ]
            - -> []
    page k@(Periodic _) = unlines $
215 [ "<h4>Periodic</h4>"
        , link ' k (html k)
        , "<h4>Periods</h4>"
        , hyper (Periods (periods k))
        ]
220 page k@(PrePeriodic _ _) = unlines $
        [ "<h4>Preperiodic</h4>"
        , link ' k (html k)
        , "<h4>Periods</h4>"
        , hyper (Periods (periods k))
225 ]

instance Page Periods where
    page (Periods (pp, p)) = unlines $
        [ "<h4>Binary Angles</h4>"
230 , "<table>" ] ++
        [ "<tr><td>" ++ hyper (BinaryAngle bb b) ++ "</td></tr>"
        | ii <- [ 0 .. bit pp - 1 ], let bb = Block ii pp
        , i <- [ 0 .. bit p - 1 ], let b = Block i p
        , binaryAngle bb b == BinaryAngle bb b
235 ] ++ [ "</table>" ] ++
        (if pp == 0 then
            [ "<h4>" ++ html Star ++ "-Periodic Kneading Sequences</h4>"
            , "<table>" ] ++
            [ "<tr><td>" ++ hyper (StarPeriodic b) ++ "</td></tr>"
240 | i <- [ 0 .. bit (p - 1) - 1 ], let b = Block i (p - 1), b ! 0
            ] ++ [ "</table>"
            , "<h4>Periodic Kneading Sequences</h4>"
            , "<table>" ] ++
            [ "<tr><td>" ++ hyper (Periodic b) ++ "</td></tr>"
245 | i <- [ 0 .. bit p - 1 ], let b = Block i p, b ! 0, b == compact b
            ] ++ [ "</table>" ]
        else

```

```

[ "<h4>Preperiodic Kneading Sequences</h4>"
, "<table>" ] ++
250 [ "<tr><td>" ++ hyper (PrePeriodic bb b) ++ "</td></tr>"
| ii <- [ 0 .. bit pp - 1 ], let bb = Block ii pp, bb ! 0
, i <- [ 0 .. bit p - 1 ], let b = Block i p
, binaryAngle bb b == BinaryAngle bb b -- fixme need generalized version
] ++ [ "</table>" ])

255 instance Hyper AngledAddress where
  hyper a
    | realizable a = "<span class='m-angledaddress'>" ++ hyper' (Unangled 1) a
    | otherwise = "<span class='m-angledaddress'>" ++ stringToHtmlString (plain
260   where
    hyper' a0 (Unangled p) = link a0 . plain $ p
    hyper' a0 (Angled p r (Unangled pp))
      = link a0 (plain p)
      ++ "<sub>" ++ link' r (html r) ++ "</sub>" ++ html ArrowRight
265   ++ link a2 (plain pp)
    where
      a2 = appendAddress a0 r pp
    hyper' a0 (Angled p r a3@(Angled pp - -))
      = link a0 (plain p)
270   ++ "<sub>" ++ link' r (html r) ++ "</sub>" ++ html ArrowRight
      ++ hyper' a2 a3
    where
      a2 = appendAddress a0 r pp

275 instance Hyper InternalAddress where
  hyper (InternalAddress (1:a)) = "<span class='m-internaladdress'>" ++ hyper' (
    InternalAddress [1]) a ++ "</span>"
  where
    hyper' a0 [] = link a0 . plain . period $ a0
    hyper' a0 (p:ps) =
280   (link a0 . plain . period $ a0) ++
    html ArrowRight ++
    hyper' (appendInternal a0 p) ps
  hyper i = "<span class='m-internaladdress'>" ++ stringToHtmlString (plain i)
    ++ "</span>"

285 appendInternal :: InternalAddress -> Int -> InternalAddress
appendInternal (InternalAddress ps) p = InternalAddress (ps ++ [p])

instance Hyper InternalAngle where
  hyper i = link' i (html i)
290 link' i s = "<a href=?q=?>" ++ q ++ "&t=internal-angle' title=?" ++ q ++ "
    (internal angle)'>" ++ s ++ "</a>"
  where q = stringToHtmlString (plain i)
instance Hyper ExternalAngle where
  hyper e = link' e (html e)
  link' e s = "<a href=?q=?>" ++ q ++ "&t=external-angle' title=?" ++ q ++ "
    (external angle)'>" ++ s ++ "</a>"
295   where q = stringToHtmlString (plain e)
instance Hyper BinaryAngle where hyper b = link' b (html b)
instance Hyper Kneading where hyper k = link' k (html k)
instance Hyper Periods where hyper p = link' p (html p)

```

```

300 appendAddress :: AngledAddress -> InternalAngle -> Int -> AngledAddress
    appendAddress (Unangled p0) r p = Angled p0 r (Unangled p)
    appendAddress (Angled p0 r0 a0) r p = Angled p0 r0 (appendAddress a0 r p)

    maxPeriod :: Int
305 maxPeriod = 100

    maxPeriodList :: Int
    maxPeriodList = 7

310 describe :: Item -> Either String (String, String, String)
    describe (BAn b)
        | preperiod b + period b > maxPeriod = Left "binary angle too large"
        | zero <= r && r <= one && binary r == b = Right (plain b, html b, "<h3>↵
            ↳ External Angle</h3>\n" ++ page b)
        | otherwise = Left "invalid binary angle"
315     where r = rational b
    describe (EAn r)
        | isNothing (safePeriods maxPeriod r) = Left "external angle too large"
        | zero <= r && r <= one = Right (plain r, html r, "<h3>External Angle</h3>\n" ↵
            ↳ ++ page (binary r))
        | otherwise = Left "invalid external angle"
320 describe (IAn i)
        | denominator i > toInteger maxPeriod = Left "internal angle too large"
        | zero < i && i < one = Right (plain i, html i, "<h3>Internal Angle</h3>\n" ++↵
            ↳ page i)
        | otherwise = Left "invalid internal angle"
    describe (Knd k)
325     | preperiod k + period k > maxPeriod = Left "kneading sequence too large"
        | otherwise = Right (plain k, html k, "<h3>Kneading Sequence</h3>\n" ++ page k↵
            ↳ )
    describe (Rat r) = Right (plain r, html r, "<h3>Rational</h3>\n" ++ if zero <= r↵
        ↳ && r <= one then "<h4>External Angle</h4>\n" ++ link ' e (html e) ++ "\n" ↵
        ↳ ++ if zero < r && r < one then "<h4>Internal Angle</h4>\n" ++ link ' i (↵
        ↳ html i) ++ "\n" else "" else "")
        where
            i = InternalAngle r
330     e = ExternalAngle r
    describe (AAd a)
        | period a > maxPeriod = Left "angled address too large"
        | realizable a = Right (plain a, html a, "<h3>Angled Address</h3>\n" ++ page a↵
            ↳ )
        | otherwise = Left "invalid angled address"
335 describe (IAd a)
        | period a > maxPeriod = Left "internal address too large"
        | otherwise = Right (plain a, html a, unlines $ [ plain a ])
    describe (Per ps@(Periods (pp, p)))
        | pp + p > maxPeriodList = Left "periods too large"
340     | pp >= 0 && p >= 1 = Right (plain ps, html ps, "<h3>Periods</h3>\n" ++ page ↵
        ↳ ps)
        | otherwise = Left "invalid periods"

app :: Application
app request respond = do
345     case lookup "q" (queryString request) of
        Just (Just b) -> case decodeUtf8' b of

```

```

Right t -> do
  let s = unpack t
  case runParser item () "" s of
350   Right i -> case describe (case i of
        Rat r -> case lookup "t" (queryString request) of
          Just (Just "internal-angle")
            | zero < r && r < one -> IAn (InternalAngle r)
          Just (Just "external-angle")
355         | zero <= r && r <= one -> EAn (ExternalAngle r)
        _ -> i
      _ -> i) of
    Left e -> err e
    Right (ptitle, title, content) ->
360     respond $ responseLBS
        status200
        [ ("Content-Type", "text/html")]
        (encodeUtf8 (pack "<html><head>\n<title>" ++ ↵
          ↳ stringToHtmlString ptitle ++ "</title>\n<style type='text↵
          ↳ /css'>\n" ++ unlines css ++ "</style></head><body><div>\n↵
          ↳ <h1>mandelbrot-web</h1>\n<h2>" ++ title ++ "</h2>\n" ++ ↵
          ↳ content ++ "\n</div></body></html>"))))
    Left e -> err $ "parse fail: " ++ show e
365   Left e -> err $ "unicode fail: " ++ show e
  _ -> err $ "not found"
where
  err s = respond $ responseLBS
        status404
370     [ ("Content-Type", "text/plain")]
        (encodeUtf8 (pack s))

css :: [String]
css =
375   [ "body { text-align: center; }"
    , "body > div { display: inline-block; margin: auto; }"
    , "body > div > dl { display: inline-block; margin: auto }"
    , "body > div > table { display: inline-block; margin: auto }"
    , "body > div > * > * { text-align: left; }"
380   , "a { text-decoration: none; }"
    , "dt { font-weight: bold; }"
    , ".m-stack { display: inline-block; text-align: right; vertical-align: middle↵
      ↳ ; }"
    , ".m-stack > * { display: block; }"
    , ".m-strong { font-weight: bold; }"
385   , ".m-binaryangle, .m-kneading { font-family: monospace; }"
    , ".m-binaryangle .m-periodic, .m-kneading .m-periodic { border-top: 1px solid↵
      ↳ ; }"
    , ".m-rational { display: inline-block; text-align: right; vertical-align: ↵
      ↳ middle; font-size: 75%; }"
    , ".m-rational .m-over { display: none; }"
    , ".m-rational .m-denominator { display: block; border-top: 1px solid; }"
390   , ".m-angledaddress .m-arrowright, .m-internaladdress .m-arrowright { margin-↵
      ↳ left: 5px; margin-right: 5px; }"
    ]

main :: IO ()
395 main = do

```

```

    putStrLn $ "http://localhost:8080/"
    run 8080 app

-- | Processing Strings into Html friendly things.
400 stringToHtmlString :: String -> String
    stringToHtmlString = concatMap fixChar
        where
            fixChar '<' = "&lt;"
            fixChar '>' = "&gt;"
405     fixChar '&' = "&amp;"
            fixChar '"' = "&quot;"
            fixChar c | ord c < 0x80 = [c]
            fixChar c = "&#" ++ show (ord c) ++ ";"

```

8 hs/client/mandelbrot.hs

```

{-# LANGUAGE FlexibleContexts #-}
{-# LANGUAGE RankNTypes #-}
{-# LANGUAGE ScopedTypeVariables #-}
module Main (main) where

5   import Prelude hiding (Rational)

    import GHCJS.Types (JSVal)

10   import GHCJS.DOM
        ( enableInspector
          , webViewGetDomDocument
          , runWebGUI
        )

15   import GHCJS.DOM.Window (getLocation , hashChange)
    import GHCJS.DOM.Document
        ( Document
          , getHead
          , getBody
20     , getElementById
          , createElement
          , createTextNode
        )

    import GHCJS.DOM.Element
25     ( setInnerHTML
          , submit
          , click
          , getBoundingClientRect
        )

30   import GHCJS.DOM.ClientRect (getLeft , getTop)
    import GHCJS.DOM.Node (appendChild)
    import GHCJS.DOM.HashChangeEvent (getNewURL)
    import GHCJS.DOM.Location (Location , getHash , setHash)
    import GHCJS.DOM.EventM (on , mouseClientXY)
35   import GHCJS.DOM.HTMLInputElement
        ( HTMLInputElement
          , castToHTMLInputElement
        )

    import GHCJS.DOM.HTMLFormElement (castToHTMLFormElement)
40   import GHCJS.DOM.HTMLInputElement
        ( HTMLInputElement

```

```

    , castToHTMLInputElement
    , getValue
    , setValue
45  )
import GHCJS.DOM.HTMLTitleElement
    ( HTMLTitleElement
    , castToHTMLTitleElement
    )
50  import GHCJS.DOM.HTMLCanvasElement
    ( castToHTMLCanvasElement
    , getContext
    )

55  import Control.Concurrent
    ( MVar
    , newEmptyMVar
    , putMVar
    , takeMVar
60  , tryTakeMVar
    , forkIOWithUnmask
    , killThread
    )

import Control.Exception (AsyncException, catch, mask_)
65  import Control.Monad (forever)
import Control.Monad.Reader (ask)
import Control.Monad.Writer (execWriter, tell)
import Control.Monad.Trans (liftIO)
import Data.Bits (bit)
70  import Data.Char (ord, chr, isHexDigit, digitToInt)
import Data.Complex (Complex((:)))
import Data.List (intersperse, nub)
import Data.Maybe (isJust)
import Text.Parsec
75  ( choice
    , try
    , getPosition
    , manyTill
    , anyChar
80  , string
    , eof
    , setPosition
    , unexpected
    , runParser
85  )
import Data.Scientific (toRealFloat)

import Mandelbrot.Symbolics
    ( AngledAddress(..)
90  , BinaryAngle(..)
    , Block(..)
    , ExternalAngle(..)
    , InternalAddress(..)
    , InternalAngle(..)
95  , Kneading(..)
    , Rational
    , Periods(..)
    , (!)

```

```

    , (%)
100    , kneading
    , associated
    , rational
    , binary
    , binaryAngle
105    , compact
    , addressAngles
    , externalAngles
    , denominator
    , zero
110    , one
    , period
    , periods
    , internalAddress
    , angledAddress
115  )
import Mandelbrot.Text
  ( Parser
    , Parse(parser)
    , Plain(plain)
120    , HTML(html)
    , Glyph(ArrowRight, Star)
  )

data View = View (Complex Double) Double
125 instance Plain View where
  plain (View (x:+y) r) = show x ++ "+" ++ show y ++ "i@" ++ show r
instance HTML View where
  html (View (x:+y) r)
    = "<span class='m-view'><span class='m-complex'>" ++
130    "<span class='m-float'>" ++ show x ++ "</span>" ++
    "<span class='m-plus'>+</span>" ++
    "<span class='m-float'>" ++ show y ++ "</span>" ++
    "<span class='m-i'>i</span></span>" ++
    "<span class='m-at'>@</span>" ++
135    "<span class='m-float'>" ++ show r ++ "</span></span>"

data Item
  = BAn BinaryAngle
  | Knd Kneading
140  | AAd AngledAddress
  | IAd InternalAddress
  | Rat Rational
  | Per Periods
  | IAn InternalAngle
145  | EAn ExternalAngle
  | Vie View

instance Parse View where
  parser = do
150    x <- parser
    string "+"
    y <- parser
    string "i"
    string "@"
155    r <- parser

```

```

    return $ View (toRealFloat x :+: toRealFloat y) (toRealFloat r)

item :: Parser Item
item = choice
160   [ t (BAn 'fmap' parser) -- starts with ., ends with )
    , t (Knd 'fmap' parser) -- starts with 0, 1, *, (, ends with )
    , t (Rat 'fmap' parser) -- contains over
    , t (Per 'fmap' parser) -- contains p
    , t (Vie 'fmap' parser) -- contains +, i and @
165   , t (AAd 'fmap' parser) -- possibly contains -, over, arrowright
    , t (IAd 'fmap' parser) -- possibly contains arrowright, no over
    , do
        p <- getPosition
        s <- manyTill anyChar eof
170        setPosition p
        unexpected . show . abbreviate 50 $ s
    ]
  where
    t p = try (do r <- p ; eof ; return r)
175    abbreviate n s = case map (splitAt n) (lines s) of
        [(s', [])] -> s'
        (s', _):-> s' ++ "..."
        _ -> ""

180
class Valid t where
    valid :: t -> Bool

instance Valid AngledAddress where
185    valid (Unangled l) = True
    valid (Angled l r' a') = go l r' a'
    where
        go p r (Unangled pp) =
            zero < r && r < one && p < pp && pp <= p * fromInteger (denominator r)
190        go p r (Angled pp r1 a1) =
            zero < r && r < one && p < pp && pp <= p * fromInteger (denominator r) ✓
            ↳ &&
            go pp r1 a1
        valid _ = False

195    realizable :: AngledAddress -> Bool
    realizable a = valid a && all (isJust . addressAngles) (truncations a)

    truncations :: AngledAddress -> [AngledAddress]
    truncations a' = execWriter $ go (Unangled 1) a'
200    where
        go a0 (Unangled _) = tell [a0]
        go a0 (Angled p r (Unangled pp)) = tell [a0,a1,a2]
        where
            a1 = appendAddress a0 r (p * fromInteger (denominator r))
205            a2 = appendAddress a0 r pp
        go a0 (Angled p r a3@(Angled pp _ _)) = tell [a0,a1] >> go a2 a3
        where
            a1 = appendAddress a0 r (p * fromInteger (denominator r))
            a2 = appendAddress a0 r pp
210
class Plain t => Hyper t where

```

```

hyper :: t -> String
link :: t -> String -> String
link t s = link' t (stringToHtmlString s)
215 link' :: t -> String -> String
link' t s = "<a href='#!q=' ++ q ++ "' title='\" ++ q ++ ">' ++ s ++ "</a>"
      where q = stringToHtmlString (plain t)

class Page t where
220 page :: t -> String

instance Page AngledAddress where
  page a0
    | realizable a0 = unlines $
225 [ "<table>" ] ++
      map tr (reverse (nub (truncations a0))) ++
      [ "</table>" ]
    | otherwise = ""
  where
230 tr a = "<tr><td>" ++ hyper a ++ "</td><td><span class='m-stack'><span>" ++
      ↪ hyper (binary lo) ++ "</span><span>" ++ hyper (binary hi) ++ "</"
      ↪ span></span></td></tr>"
    where
      Just (lo, hi) = addressAngles a

instance Page BinaryAngle where
235 page b = unlines $
  [ "<h4>Binary</h4>"
    , link' b (html b)
    , "<h4>Rational</h4>"
    , link' r (html r)
240 , "<h4>Periods</h4>"
    , hyper (Periods (periods b))
    , "<h4>Kneading</h4>"
    , link' k (html k)
    , case i' of Just i -> "<h4>Internal Address</h4>\n" ++ link' i (html i) ; ↪
      ↪ Nothing -> ""
245 , case a' of Just a -> "<h4>Angled Address</h4>\n" ++ link' a (html a) ; ↪
      ↪ Nothing -> ""
  ] ++ if null (drop 1 es) then [] else
    [ "<h4>Landings</h4>"
      , "<table>" ] ++
      map tr es ++
250 [ "</table>" ]
  where
    r = rational b
    k = kneading r
    i' = internalAddress k
    a' = angledAddress r
255 es = externalAngles b
    tr e = "<tr>" ++ (if b == e then " class='m-strong'" else "") ++ "><td>" ++
      ↪ hyper e ++ "</td><td>" ++ hyper (rational e) ++ "</td></tr>"

instance Page InternalAngle where
260 page i = unlines $
  [ "<h4>Rational</h4>"
    , link' i (html i)
    , "<h4>Bulb</h4>"

```

```

    , "<table>" ] ++
265   map tr [lo, hi] ++
    [ "</table>"
{-
    , "<h4>Spokes</h4>"
    , "<table>"
270   , map tr spokes
    , "</table>"
-}
  ]
  where
275   Just (lo, hi) = addressAngles (Angled 1 i (Unangled (fromInteger (↯
    ↯ denominator i))))
    tr e = "<tr><td>" ++ hyper (binary e) ++ "</td><td>" ++ hyper e ++ "</td>↯
    ↯ ></tr>"

instance Page Kneading where
  page k@(StarPeriodic _) = unlines $
280   [ "<h4>" ++ html Star ++ "-Periodic</h4>"
    , link ' k (html k)
    , "<h4>Periods</h4>"
    , hyper (Periods (periods k))
  ] ++ case associated k of
285   Just (u, l) ->
    [ "<h4>Upper</h4>"
    , link ' u (html u)
    , "<h4>Lower</h4>"
    , link ' l (html l)
290   ]
    - -> []
  page k@(Periodic _) = unlines $
    [ "<h4>Periodic</h4>"
    , link ' k (html k)
295   , "<h4>Periods</h4>"
    , hyper (Periods (periods k))
  ]
  page k@(PrePeriodic _ _) = unlines $
    [ "<h4>Preperiodic</h4>"
    , link ' k (html k)
300   , "<h4>Periods</h4>"
    , hyper (Periods (periods k))
  ]

305 instance Page Periods where
  page (Periods (pp, p)) = unlines $
    [ "<h4>Binary Angles</h4>"
    , "<table>" ] ++
    [ "<tr><td>" ++ hyper (BinaryAngle bb b) ++ "</td></tr>"
310   | ii <- [ 0 .. bit pp - 1 ], let bb = Block ii pp
    , i <- [ 0 .. bit p - 1 ], let b = Block i p
    , binaryAngle bb b == BinaryAngle bb b
  ] ++ [ "</table>" ] ++
    (if pp == 0 then
315   [ "<h4>" ++ html Star ++ "-Periodic Kneading Sequences</h4>"
    , "<table>" ] ++
    [ "<tr><td>" ++ hyper (StarPeriodic b) ++ "</td></tr>"
    | i <- [ 0 .. bit (p - 1) - 1 ], let b = Block i (p - 1), b ! 0

```

```

    ] ++ [ "</table>"
320   , "<h4>Periodic Kneading Sequences</h4>"
    , "<table>" ] ++
    [ "<tr><td>" ++ hyper (Periodic b) ++ "</td></tr>"
    | i <- [ 0 .. bit p - 1 ], let b = Block i p, b ! 0, b == compact b
    ] ++ [ "</table>" ]
325   else
    [ "<h4>Preperiodic Kneading Sequences</h4>"
    , "<table>" ] ++
    [ "<tr><td>" ++ hyper (PrePeriodic bb b) ++ "</td></tr>"
    | ii <- [ 0 .. bit pp - 1 ], let bb = Block ii pp, bb ! 0
330   , i <- [ 0 .. bit p - 1 ], let b = Block i p
    , binaryAngle bb b == BinaryAngle bb b -- fixme need generalized version
    ] ++ [ "</table>" ])

instance Page View where
335   page _ = unlines $
    [ "<canvas id='canvas' width='1024' height='576'></canvas>" ]

instance Hyper AngledAddress where
  hyper a
340   | realizable a = "<span class='m-angledaddress'>" ++ hyper' (Unangled 1) a
    ↪ ++ "</span>"
    | otherwise = "<span class='m-angledaddress'>" ++ stringToHtmlString (plain ↪
    ↪ a) ++ "</span>"
    where
      hyper' a0 (Unangled p) = link a0 . plain $ p
      hyper' a0 (Angled p r (Unangled pp))
345       = link a0 (plain p)
        ++ "<sub>" ++ link' r (html r) ++ "</sub>" ++ html ArrowRight
        ++ link a2 (plain pp)
        where
          a2 = appendAddress a0 r pp
350   hyper' a0 (Angled p r a3@(Angled pp _))
    = link a0 (plain p)
    ++ "<sub>" ++ link' r (html r) ++ "</sub>" ++ html ArrowRight
    ++ hyper' a2 a3
    where
355       a2 = appendAddress a0 r pp

instance Hyper InternalAddress where
  hyper (InternalAddress (1:a)) = "<span class='m-internaladdress'>" ++ hyper' (↪
    ↪ InternalAddress [1]) a ++ "</span>"
    where
360   hyper' a0 [] = link a0 . plain . period $ a0
    hyper' a0 (p:ps) =
      (link a0 . plain . period $ a0) ++
      html ArrowRight ++
      hyper' (appendInternal a0 p) ps
365   hyper i = "<span class='m-internaladdress'>" ++ stringToHtmlString (plain i) ↪
    ↪ ++ "</span>"

appendInternal :: InternalAddress -> Int -> InternalAddress
appendInternal (InternalAddress ps) p = InternalAddress (ps ++ [p])

370 instance Hyper InternalAngle where
  hyper i = link' i (html i)

```

```

    link ' i s = "<a href='#!q=' ++ q ++ "&t=internal-angle' title='" ++ q ++ "↵
      ↳ (internal angle)'>" ++ s ++ "</a>"
    where q = stringToHtmlString (plain i)
instance Hyper ExternalAngle where
375   hyper e = link ' e (html e)
    link ' e s = "<a href='#!q=' ++ q ++ "&t=external-angle' title='" ++ q ++ "↵
      ↳ (external angle)'>" ++ s ++ "</a>"
    where q = stringToHtmlString (plain e)
instance Hyper BinaryAngle where hyper b = link ' b (html b)
instance Hyper Kneading where hyper k = link ' k (html k)
380 instance Hyper Periods where hyper p = link ' p (html p)
instance Hyper View where hyper v = link ' v (html v)

appendAddress :: AngledAddress -> InternalAngle -> Int -> AngledAddress
appendAddress (Unangled p0) r p = Angled p0 r (Unangled p)
385 appendAddress (Angled p0 r0 a0) r p = Angled p0 r0 (appendAddress a0 r p)

maxPeriodList :: Int
maxPeriodList = 10

390 describe :: Item -> Either String (String, String, String)
describe (BAN b)
  | zero <= r && r <= one = Right (plain b', html b', "<h3>External Angle</h3>\n"
    ↳ " ++ page b')
  | otherwise = Left "invalid binary angle"
  where
395   r = rational b
   b' = binary r
describe (EAN r)
  | zero <= r && r <= one = Right (plain r, html r, "<h3>External Angle</h3>\n"
    ↳ ++ page (binary r))
  | otherwise = Left "invalid external angle"
400 describe (IAN i)
  | zero < i && i < one = Right (plain i, html i, "<h3>Internal Angle</h3>\n" ++
    ↳ page i)
  | otherwise = Left "invalid internal angle"
describe (Knd k)
  | otherwise = Right (plain k, html k, "<h3>Kneading Sequence</h3>\n" ++ page k
    ↳ )
405 describe (Rat r) = Right (plain r, html r, "<h3>Rational</h3>\n" ++ if zero <= r
    ↳ && r <= one then "<h4>External Angle</h4>\n" ++ link ' e (html e) ++ "\n"
    ↳ ++ if zero < r && r < one then "<h4>Internal Angle</h4>\n" ++ link ' i (
    ↳ html i) ++ "\n" else "" else "")
  where
    i = InternalAngle r
    e = ExternalAngle r
describe (AAd a)
410   | realizable a = Right (plain a, html a, "<h3>Angled Address</h3>\n" ++ page a
    ↳ )
    | otherwise = Left "invalid angled address"
describe (IAd a)
  | otherwise = Right (plain a, html a, unlines $ [ plain a ])
describe (Per ps@(Periods (pp, p)))
415   | pp + p > maxPeriodList = Left $ "periods too large, period limit " ++ show
    ↳ maxPeriodList
    | pp >= 0 && p >= 1 = Right (plain ps, html ps, "<h3>Periods</h3>\n" ++ page
    ↳ ps)

```

```

    | otherwise = Left "invalid periods"
describe (Vie v) = Right (plain v, html v, "<h3>View</h3>" ++ page v)

420 help :: [String]
help = [ "<h2>Examples</h2>" ] ++ intersperse ", "
    [ hyper (BinaryAngle (Block 0 0) (Block 1 2))
    , hyper (BinaryAngle (Block 2 2) (Block 1 2))
    , hyper (2%5 :: ExternalAngle)
425   , hyper (2%5 :: InternalAngle)
    , hyper (StarPeriodic (Block 2 2))
    , hyper (Periodic (Block 2 2))
    , hyper (PrePeriodic (Block 1 2) (Block 2 2))
    , link ' a (html a)
430   , link ' i (html i)
    , hyper (Periods (0, 4))
    , hyper (Periods (2, 2))
    , hyper (View ((-0.75) :+ 0) 1.25)
    ]
435 where
    a = Angled 1 (1%2) (Angled 2 (1%3) (Unangled 5))
    i = InternalAddress [1,2,5]

worker :: Location -> Document -> HTMLInputElement -> HTMLTitleElement -> ↵
    ↵ HTMLInputElement -> MVar String -> (forall a. IO a -> IO a) -> IO ()
440 worker loc doc input titleNode output mv unmask =
    forever $ unmask update `catch` \(e :: AsyncException) -> return ()
    where
        update = do
            url <- takeMVar mv
445         setInnerHTML output (Just $ "<span class='busy'>...calculating...</span ↵
            ↵ >")
        case dropWhile (/= '#') (percentDecode url) of
            '#':':!':s -> case fmap (runParser item () "") (lookup "q" (parseAnchor ↵
            ↵ s)) of
                Just (Right i) -> case describe (case i of
                    Rat r -> case lookup "t" (parseAnchor s) of
450                     Just "internal-angle"
                        | zero < r && r < one -> IAn (InternalAngle r)
                        Just "external-angle"
                        | zero <= r && r <= one -> EAn (ExternalAngle r)
                        - -> i
455                     - -> i) of
                    Left e -> setInnerHTML output (Just $ "<span class='error'>" ++ ↵
                    ↵ stringToHtmlString (show e) ++ "</span>" ++ unlines help)
                    Right (ptitle, title, content) -> do
                        setValue input (lookup "q" (parseAnchor s))
                        setInnerHTML titleNode (Just $ stringToHtmlString ptitle)
460                     setInnerHTML output (Just $ "<h2>" ++ title ++ "</h2>\n" ++ ↵
                    ↵ content ++ "\n")
                        case i of
                            Vie (View (cx:cy) r) -> do
                                Just canvas <- fmap castToHTMLCanvasElement <$> ↵
                                ↵ getElementById doc "canvas"
                                context <- getContext canvas "2d"
465                             js_m_d_render context 1024 576 cx cy r 25 100000
                                on canvas click $ do
                                    Just rect <- getBoundingClientRect canvas

```

```

470      rx <- realToFrac <$> getLeft rect
      ry <- realToFrac <$> getTop rect
      (ix, iy) <- mouseClientXY
      let dx = 2 * ((fromIntegral ix + 0.5 - rx) / 1024 - 0.5) * 2
          ↪ (1024 / 576)
          dy = 2 * (0.5 - (fromIntegral iy + 0.5 - ry) / 576)
          ddx = r * dx / 2
          ddy = r * dy / 2
475      cx' = cx + ddx
      cy' = cy + ddy
      r' = r / 2
      setHash loc $ "#!q=" ++ plain (View (cx' :+ cy') r')
      return ()
480      return ()
      - -> return ()
      Just (Left e) -> setInnerHTML output (Just $ "<span class='error'>↵
          ↪ parse failure: " ++ stringToHtmlString (show e) ++ "</span>" ↵
          ↪ ++ unlines help)
      - -> setInnerHTML output (Just $ unlines help)
      - -> setInnerHTML output (Just $ unlines help)
485
main :: IO ()
main = runWebGUI $ \window -> do
  enableInspector window
  Just loc <- getLocation window
490  Just doc <- webViewGetDomDocument window
  Just head' <- getHead doc
  Just titleNode <- fmap castToHTMLTitleElement <$> createElement doc (Just "↵
      ↪ title")
  ttext <- createTextNode doc $ "mandelbrot-web"
  appendChild titleNode ttext
495  appendChild head' (Just titleNode)
  Just style <- createElement doc (Just "style")
  text <- createTextNode doc $ unlines css
  appendChild style text
  appendChild head' (Just style)
500  Just body <- getBody doc
  setInnerHTML body (Just $ "<h1>mandelbrot-web</h1><form id='form'><input id='↵
      ↪ input' size='60' /><button type='submit'>Submit</button></form><div id='↵
      ↪ output'>" ++ unlines help ++ "</div>")
  Just form <- fmap castToHTMLFormElement <$> getElementById doc "form"
  Just input <- fmap castToHTMLInputElement <$> getElementById doc "input"
  Just output <- fmap castToHTMLElement <$> getElementById doc "output"
505  mv <- newEmptyMVar
  workerThread <- mask_ $ forkIOWithUnmask (worker loc doc input titleNode ↵
      ↪ output mv)
  on window hashChange $ do
    url <- getNewURL ==<< ask
    liftIO $ do
510      killThread workerThread
      tryTakeMVar mv -- discard existing
      putMVar mv url
    return ()
  on form submit $ do
515  Just s <- getValue input
    setHash loc $ "#!q=" ++ s
    return ()

```

```

    url <- getHash loc
    putMVar mv url
520    return ()

parseAnchor :: String -> [(String, String)]
parseAnchor "" = []
parseAnchor s = case break (== '&') s of
525   (q, r) -> case break (== '=') q of
        (a, b) -> (a, drop 1 b) : parseAnchor (drop 1 r)

css :: [String]
css =
530   [ "body { text-align: center; }"
     , "body > div#output { display: inline-block; margin: auto; }"
     , "body > div#output > dl { display: inline-block; margin: auto }"
     , "body > div#output > table { display: inline-block; margin: auto }"
     , "body > div#output > * > * { text-align: left; }"
535   , "a { text-decoration: none; }"
     , "dt { font-weight: bold; }"
     , ".m-stack { display: inline-block; text-align: right; vertical-align: middle;
        ↵ ; }"
     , ".m-stack > * { display: block; }"
     , ".m-strong { font-weight: bold; }"
540   , ".m-binaryangle, .m-kneading, .m-view { font-family: monospace; }"
     , ".m-binaryangle .m-periodic, .m-kneading .m-periodic { border-top: 1px solid;
        ↵ ; }"
     , ".m-rational { display: inline-block; text-align: right; vertical-align: ↵
        ↵ middle; font-size: 75%; }"
     , ".m-rational .m-over { display: none; }"
     , ".m-rational .m-denominator { display: block; border-top: 1px solid; }"
545   , ".m-angledaddress .m-arrowright, .m-internaladdress .m-arrowright { margin-↵
        ↵ left: 5px; margin-right: 5px; }"
     , ".m-complex .m-plus, .m-complex .m-i, .m-view .m-at { font-weight: bold; }"
     , ".error { background-color: #ffdddd; }"
     , ".busy { background-color: #ffeedd; }"
     , "canvas { border: 1px solid; }"
550   ]

-- | Processing Strings into Html friendly things.
stringToHtmlString :: String -> String
stringToHtmlString = concatMap fixChar
555   where
        fixChar '<' = "&lt;"
        fixChar '>' = "&gt;"
        fixChar '&' = "&amp;"
        fixChar '"' = "&quot;"
560        fixChar c | ord c < 0x80 = [c]
        fixChar c = "&#" ++ show (ord c) ++ ";"

-- from happstack-server 7.4.6.1
percentDecode :: String -> String
565 percentDecode [] = ""
percentDecode ('%':x1:x2:s) | isHexDigit x1 && isHexDigit x2 =
    chr (digitToInt x1 * 16 + digitToInt x2) : percentDecode s
percentDecode (c:s) = c : percentDecode s

570 foreign import javascript "m_d-render($1,$2,$3,$4,$5,$6,$7,$8);" js_m_d_render ↵

```

```
↳ :: JSVal -> Int -> Int -> Double -> Double -> Double -> Double -> Int -> ↵
↳ IO ()
```

9 js/m-render.html

```
<!DOCTYPE html>
<html><head><script src="m-render.js"></script></head>
<body style="margin: auto; width: 1024px;">
<h1>mandelbrot</h1>
5 <form id="view">
  <input id="centerx" size="20" value="-0.75"></input> +
  <input id="centery" size="20" value="0"></input> i
  <input id="radius" size="20" value="1.5"></input> @
  <button id="go">Go</button>
10 </form>
  <canvas id="canvas" width="1024" height="576" style="border: 1px solid;"></↵
    ↳ canvas></body>
<script defer>
function doit() {
  var cx = parseFloat(document.getElementById("centerx").value);
15  var cy = parseFloat(document.getElementById("centery").value);
  var r = parseFloat(document.getElementById("radius").value);
  var cnv = document.getElementById("canvas");
  var ctx = cnv.getContext("2d");
  var img = m_d_render(ctx, 1024, 576, cx, cy, r, 25, 10000);
20  ctx.putImageData(img, 0, 0);
}
var view = document.getElementById("view");
view.onsubmit = doit;
doit();
25 </script>
</html>
```

10 js/m-render.js

```
function m_d_wucleus_step(out, z_guessx, z_guessy, cx, cy, period) {
  var epsilon2 = 1e-24;
  var zzx = z_guessx;
  var zzy = z_guessy;
5  var dzzx = 1;
  var dzzy = 0;
  var t;
  var i;
  for (i = 0; i < period; ++i) {
10    // dzz = 2 * zz * dzz;
    t = 2 * (zzx * dzzx - zzy * dzzy);
    dzzy = 2 * (zzx * dzzy + zzy * dzzx);
    dzzx = t;
    // zz = zz * zz + c;
15    t = zzx * zzx - zzy * zzy + cx;
    zzy = 2 * zzx * zzy + cy;
    zzx = t;
  }
  var dx = zzx - z_guessx;
20  var dy = zzy - z_guessy;
  if (dx * dx + dy * dy <= epsilon2) {
```

```

        out.zx = z_guessx;
        out.zy = z_guessy;
        return "converged";
25    }
    // z_new = z_guess - (zz - z_guess) / (dzz - 1);
    dzzx = dzzx - 1;
    var dzz = dzzx * dzzx + dzzz * dzzz;
    var z_newx = z_guessx - (dx * dzzx + dy * dzzz) / dzz;
30    var z_newy = z_guessy - (dy * dzzx - dx * dzzz) / dzz;
    // d = z_new - zz;
    dx = z_newx - z_guessx;
    dy = z_newy - z_guessy;
    if (dx * dx + dy * dy <= epsilon2) {
35        out.zx = z_newx;
        out.zy = z_newy;
        return "converged";
    }
    if (dx * dx + dy * dy < 1.0/0.0) {
40        out.zx = z_newx;
        out.zy = z_newy;
        return "stepped";
    } else {
        out.zx = z_guessx;
45        out.zy = z_guessy;
        return "failed";
    }
}

50 function m_d_wucleus(out, z_guessx, z_guessy, cx, cy, period, maxsteps) {
    var result = "failed";
    var z = { zx: z_guessx, zy: z_guessy };
    var i;
    for (i = 0; i < maxsteps; ++i) {
55        if ("stepped" != (result = m_d_wucleus_step(z, z.zx, z.zy, cx, cy, period))) ↯
            ↵ {
                break;
            }
    }
    out.zx = z.zx;
60    out.zy = z.zy;
    return result;
}

function m_d_interior_de(out, zx, zy, cx, cy, p, steps) {
65    var out2 = { zx: 0, zy: 0 };
    if ("failed" != m_d_wucleus(out2, zx, zy, cx, cy, p, steps)) {
        var z0x = out2.zx;
        var z0y = out2.zy;
        var dz0x = 1;
70        var dz0y = 0;
        var t;
        var j;
        for (j = 0; j < p; ++j) {
            // dz0 = 2 * z0 * dz0;
75            t = 2 * (z0x * dz0x - z0y * dz0y);
            dz0y = 2 * (z0x * dz0y + z0y * dz0x);
            dz0x = t;

```

```

    // z0 = z0 * z0 + c;
    t = z0x * z0x - z0y * z0y + cx;
80    z0y = 2 * z0x * z0y + cy;
    z0x = t;
}
if (dz0x * dz0x + dz0y * dz0y <= 1) {
85    var z1x = out2.zx;
    var z1y = out2.zy;
    var dz1x = 1;
    var dz1y = 0;
    var dcdz1x = 0;
    var dcdz1y = 0;
90    var dclx = 0;
    var dcly = 0;
    var dcdz1x = 0;
    var dcdz1y = 0;
    for (j = 0; j < p; ++j) {
95        // dcdz1 = 2 * (z1 * dcdz1 + dz1 * dcl);
        t = 2 * (z1x * dcdz1x - z1y * dcdz1y + dz1x * dclx - dz1y * dcly);
        dcdz1y = 2 * (z1x * dcdz1y + z1y * dcdz1x + dz1x * dcly + dz1y * dclx);
        dcdz1x = t;
        // dcl = 2 * z1 * dcl + 1;
100        t = 2 * (z1x * dclx - z1y * dcly) + 1;
        dcly = 2 * (z1x * dcly + z1y * dclx);
        dclx = t;
        // dzdz1 = 2 * (dz1 * dz1 + z1 * dzdz1);
        t = 2 * (dz1x * dz1x - dz1y * dz1y + z1x * dzdz1x - z1y * dzdz1y);
105        dzdz1y = 2 * (2 * dz1x * dz1y + z1x * dzdz1y + z1y * dzdz1x);
        dzdz1x = t;
        // dz1 = 2 * z1 * dz1;
        t = 2 * (z1x * dz1x - z1y * dz1y);
        dz1y = 2 * (z1x * dz1y + z1y * dz1x);
110        dz1x = t;
        // z1 = z1 * z1 + c;
        t = z1x * z1x - z1y * z1y + cx;
        z1y = 2 * z1x * z1y + cy;
        z1x = t;
115    }
    out.dzx = dz1x;
    out.dzy = dz1y;
    // *de_out = (1 - cabs2(dz1)) / cabs(dcdz1 + dzdz1 * dcl / (1 - dz1));
    var bx = 1 - dz1x;
120    var by = -dz1y;
    var b = bx * bx + by * by;
    var ax = dzdz1x * dclx - dzdz1y * dcly;
    var ay = dzdz1x * dcly + dzdz1y * dclx;
    t = (ax * bx + ay * by) / b + dcdz1x;
125    ay = (ay * bx - ax * by) / b + dcdz1y;
    ax = t;
    out.de = (1 - dz1x * dz1x - dz1y * dz1y) / Math.sqrt(ax * ax + ay * ay);
    return true;
}
130 }
    return false;
}

function m_d_compute_init(px, bias, er, cx, cy) {

```

```

135     px.tag = "unknown";
        px.bias = bias;
        px.er2 = er * er;
        px.mz2 = 1.0 / 0.0;
        px.cx = cx;
140     px.cy = cy;
        px.zx = 0;
        px.zy = 0;
        px.dcx = 0;
        px.dcy = 0;
145     px.dzx = 0;
        px.dzy = 0;
        px.n = 0;
        px.p = 0;
        px.np = 0;
150     px.partials = [];
        px.de = -1;
    }

    function m_d.compute_clear(px) {
155     px.tag = "unknown";
        px.np = 0;
        px.partials = [];
    }

160     function m_d.compute_step(px, steps) {
        if (px.tag != "unknown") {
            return true;
        }
        var er2 = px.er2;
165     var cx = px.cx;
        var cy = px.cy;
        var zx = px.zx;
        var zy = px.zy;
        var dcx = px.dcx;
170     var dcy = px.dcy;
        var mz2 = px.mz2;
        var p = px.p;
        var t;
        var z2;
175     var i;
        for (i = 1; i <= steps; ++i) {
            // dc = 2 * z * dc + 1;
            t = 2 * (zx * dcx - zy * dcy) + 1;
            dcy = 2 * (zx * dcy + zy * dcx);
180     dcx = t;
            // z = z * z + c;
            t = zx * zx - zy * zy + cx;
            zy = 2 * zx * zy + cy;
            zx = t;
185     // z2 = cabs2(z);
            z2 = zx * zx + zy * zy;
            if (z2 < mz2) {
                mz2 = z2;
                p = px.n + i;
190     if (px.bias == "interior") {
                var out = { dzx: 0, dzy: 0, de: -1 };

```

```

        if (m_d_interior_de(out, zx, zy, cx, cy, p, 64)) {
            px.tag = "interior";
            px.p = p;
195         px.zx = zx;
            px.zy = zy;
            px.dzx = out.dzx;
            px.dzy = out.dzy;
            px.de = out.de;
200         return true;
        }
    } else {
        px.partials[px.np] = { zx: zx, zy: zy, p: p };
        px.np = px.np + 1;
205     }
    }
    if (! (z2 < er2)) {
        px.tag = "exterior";
        px.n = px.n + i;
210         px.p = p;
        px.zx = zx;
        px.zy = zy;
        px.dcx = dcx;
        px.dcy = dcy;
215         px.de = Math.sqrt(z2 / (dcx * dcx + dcy * dcy)) * Math.log(z2);
        return true;
    }
}
if (px.bias != "interior") {
220     for (i = 0; i < px.np; ++i) {
        zx = px.partials[i].zx;
        zy = px.partials[i].zy;
        p = px.partials[i].p;
        var out = { dzx: 0, dzy: 0, de: -1 };
225         if (m_d_interior_de(out, zx, zy, cx, cy, p, 64)) {
            px.tag = "interior";
            px.p = p;
            px.zx = zx;
            px.zy = zy;
230             px.dzx = out.dzx;
            px.dzy = out.dzy;
            px.de = out.de;
            return true;
        }
235     }
}
px.tag = "unknown";
px.n = px.n + steps;
px.p = p;
240 px.mz2 = mz2;
px.zx = zx;
px.zy = zy;
px.dcx = dcx;
px.dcy = dcy;
245 return false;
}

function m_d_render(ctx, width, height, centerx, centery, r, er, maxiters) {

```

```

var image = ctx.createImageData(width, height);
250 var imagedata = image.data;
var k = 0;
var j;
var i;
var px = { tag: "unknown" };
255 for (j = 0; j < height; ++j) {
    for (i = 0; i < width; ++i) {
        var cx = centerx + r * (i - (width/2 - 0.5)) / (height/2);
        var cy = centery - r * (j - (height/2 - 0.5)) / (height/2);
        var dc = r / (height/2);
260 m_d.compute_init(px, px.tag, er, cx, cy);
        m_d.compute_step(px, maxiters);
        var g = 0;
        if (px.tag == "exterior" || px.tag == "interior") {
            g = Math.round(255 * Math.tanh(Math.min(Math.max(px.de / dc, 0), 8)));
265 }
        imagedata[k++] = g;
        imagedata[k++] = g;
        imagedata[k++] = g;
        imagedata[k++] = 255;
270 }
    }
}
ctx.putImageData(image, 0, 0);
}

```

11 mandelbrot-web.cabal

```

name:                mandelbrot-web
version:             0.1.0.0
synopsis:             web interface related to the Mandelbrot set
description:         Web interface related to the Mandelbrot set.
5 homepage:          https://code.mathr.co.uk/mandelbrot-web
license:             GPL-3
license-file:        COPYING
author:              Claude Heiland-Allen
maintainer:          claude@mathr.co.uk
10 copyright:         (c) 2015 Claude Heiland-Allen
category:            Math
build-type:          Simple
cabal-version:       >=1.10
extra-source-files:
15   README.md

executable mandelbrot
  main-is: mandelbrot.hs
  build-depends:
20   base          >=4.7 && <4.14,
   ghcjs-base     >=0.2 && <0.3,
   ghcjs-dom      >=0.2 && <0.3,
   text           >=1.2 && <1.3,
   parsec         >=3.1 && <3.2,
25   mtl           >=2.1 && <2.3,
   scientific     >=0.3 && <0.4,
   mandelbrot-symbolics >=0.1 && <0.2,
   mandelbrot-text >=0.1 && <0.2
  hs-source-dirs:    hs/client

```

```
30  default-language:      Haskell2010
    other-extensions:
        JavaScriptFFI
        FlexibleContexts
    ghc-options:           -Wall -fno-warn-unused-do-bind -DGHCIJS_BROWSER
35  ghc-prof-options:      -prof -auto-all -caf-all

source-repository head
  type:      git
  location:  https://code.mathr.co.uk/mandelbrot-web.git
40
source-repository this
  type:      git
  location:  https://code.mathr.co.uk/mandelbrot-web.git
  tag:       v0.1.0.0
```

12 README.md

mandelbrot-web

Web interface related to the Mandelbrot set.

13 Setup.hs

```
import Distribution.Simple
main = defaultMain
```