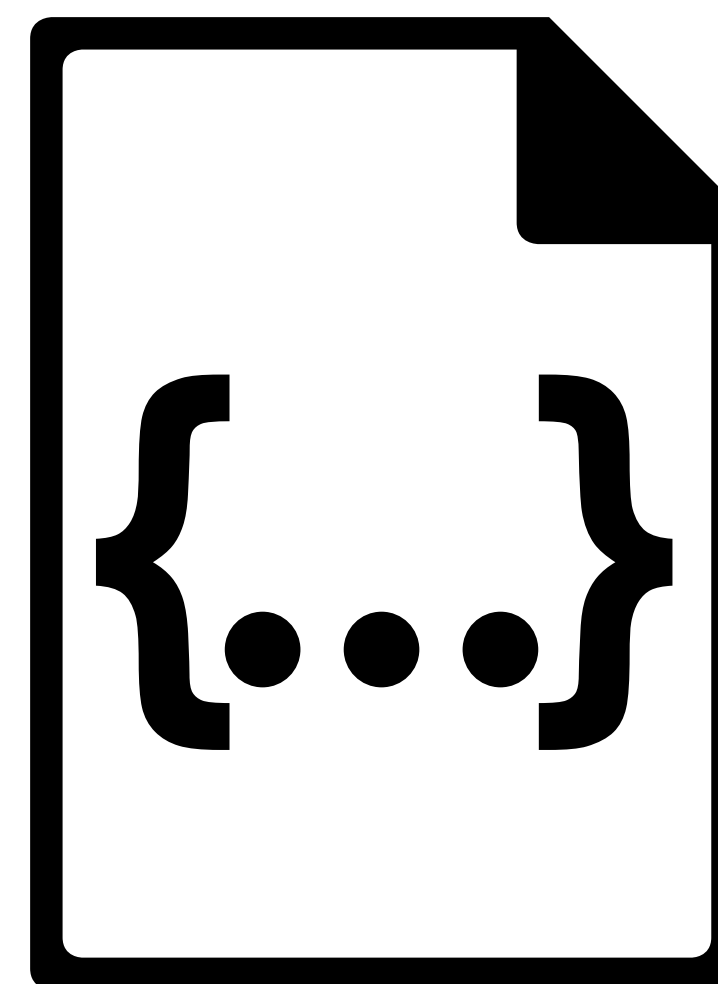
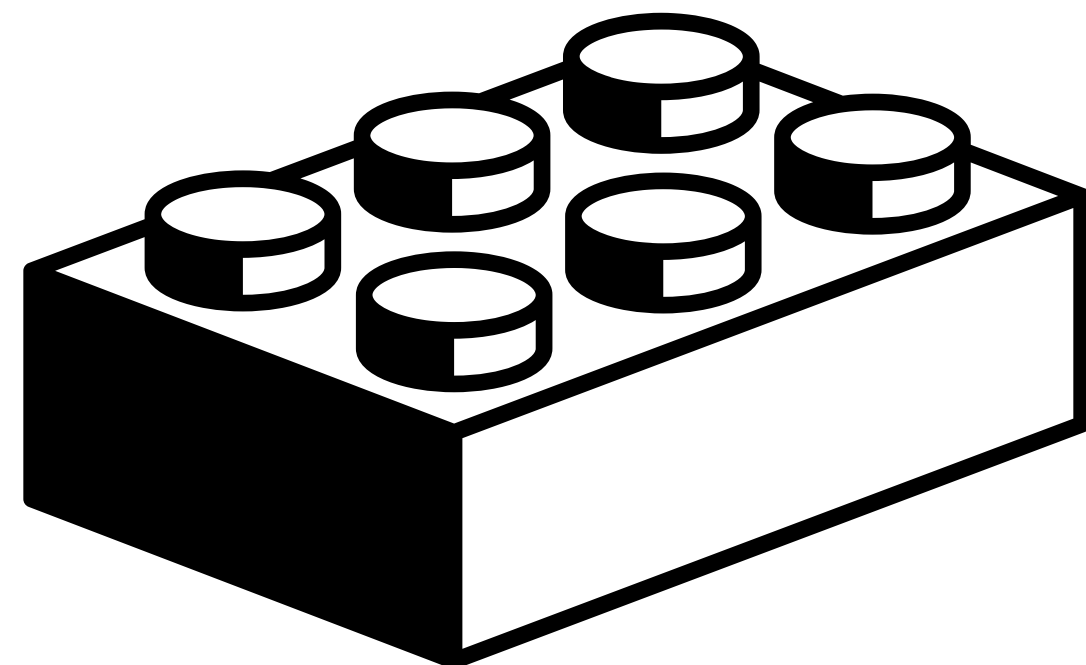




CS 4241

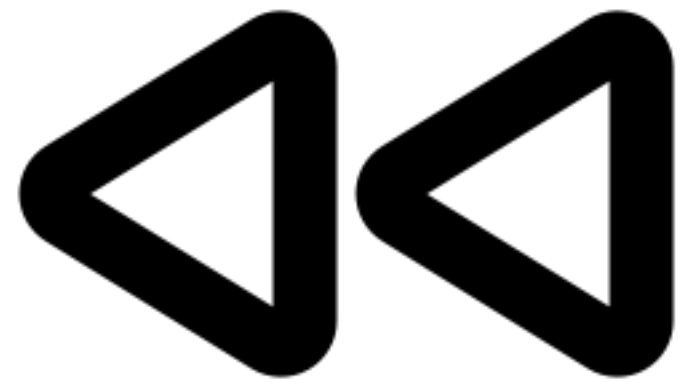
WebWare

basics, html, css



basics, html, css



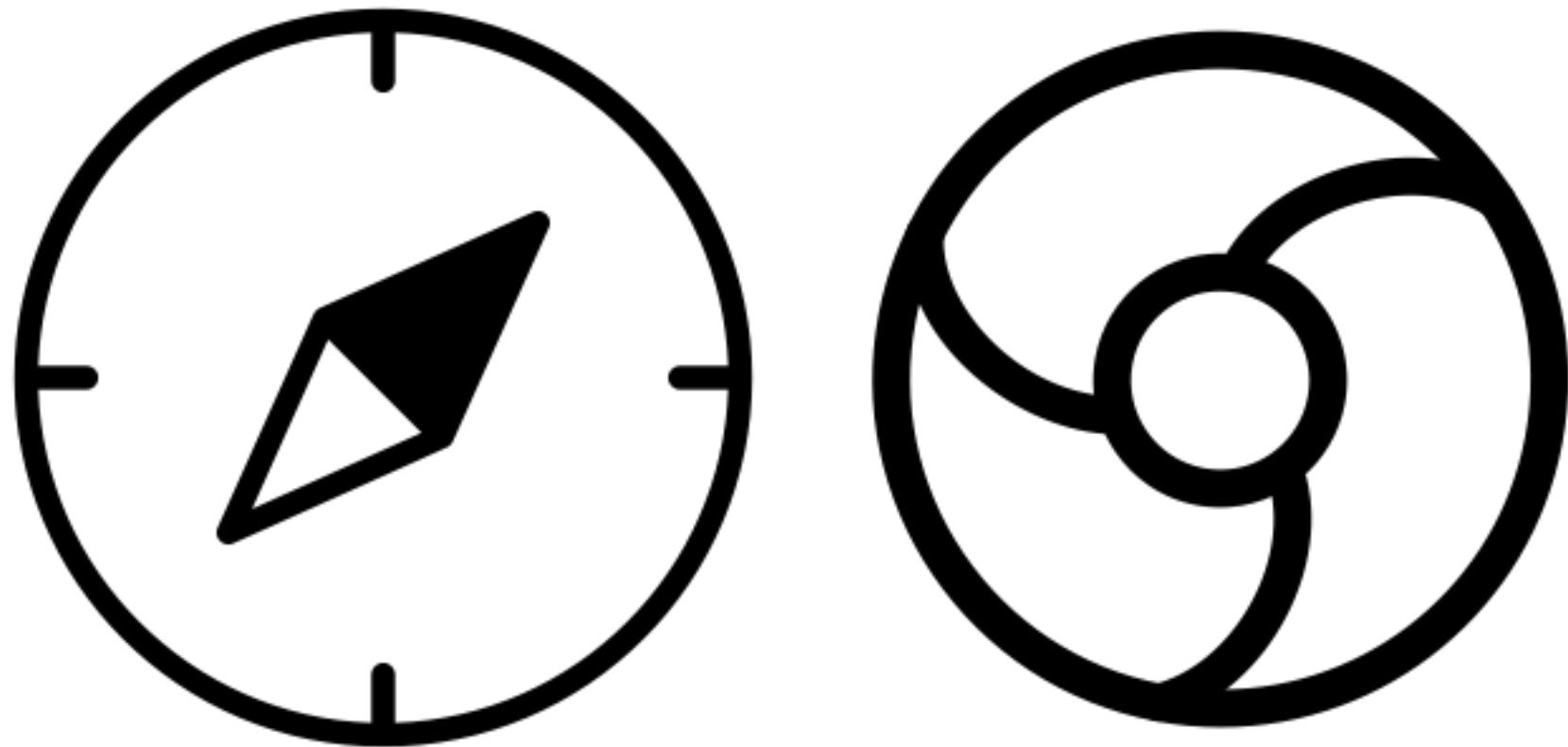


color should change



```
0 <script src="https://d3js.org/d3.v4.min.js"></script>
1
2 <div id="name">color should change</div>
3 <svg id="vis" width=400 height=400></svg>
4
5 <script>
6 console.log(d3)
7
8 d3.csv('data.csv', function(err, d) {
9
```

browsers

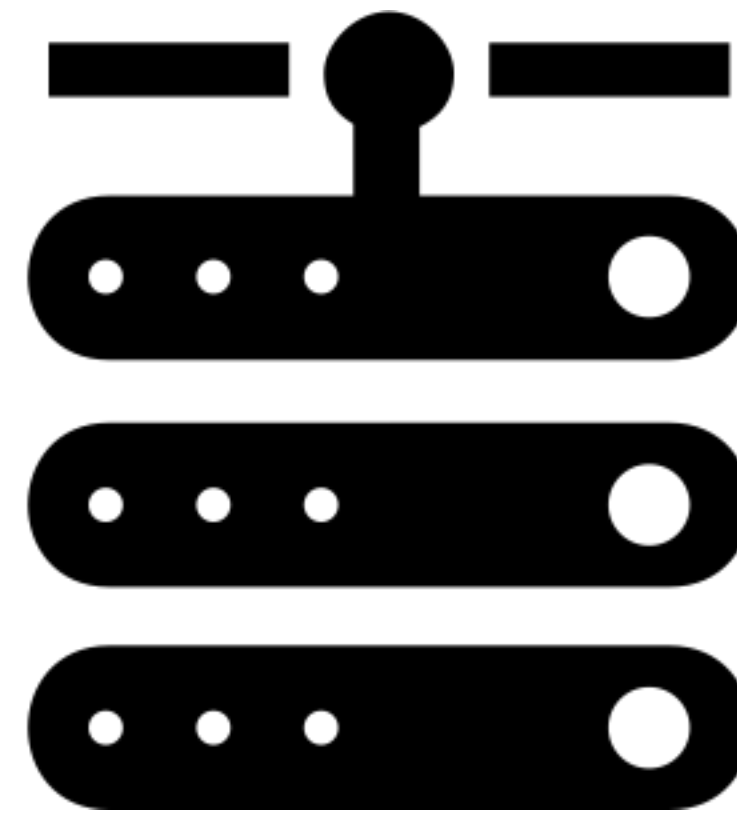


front-end

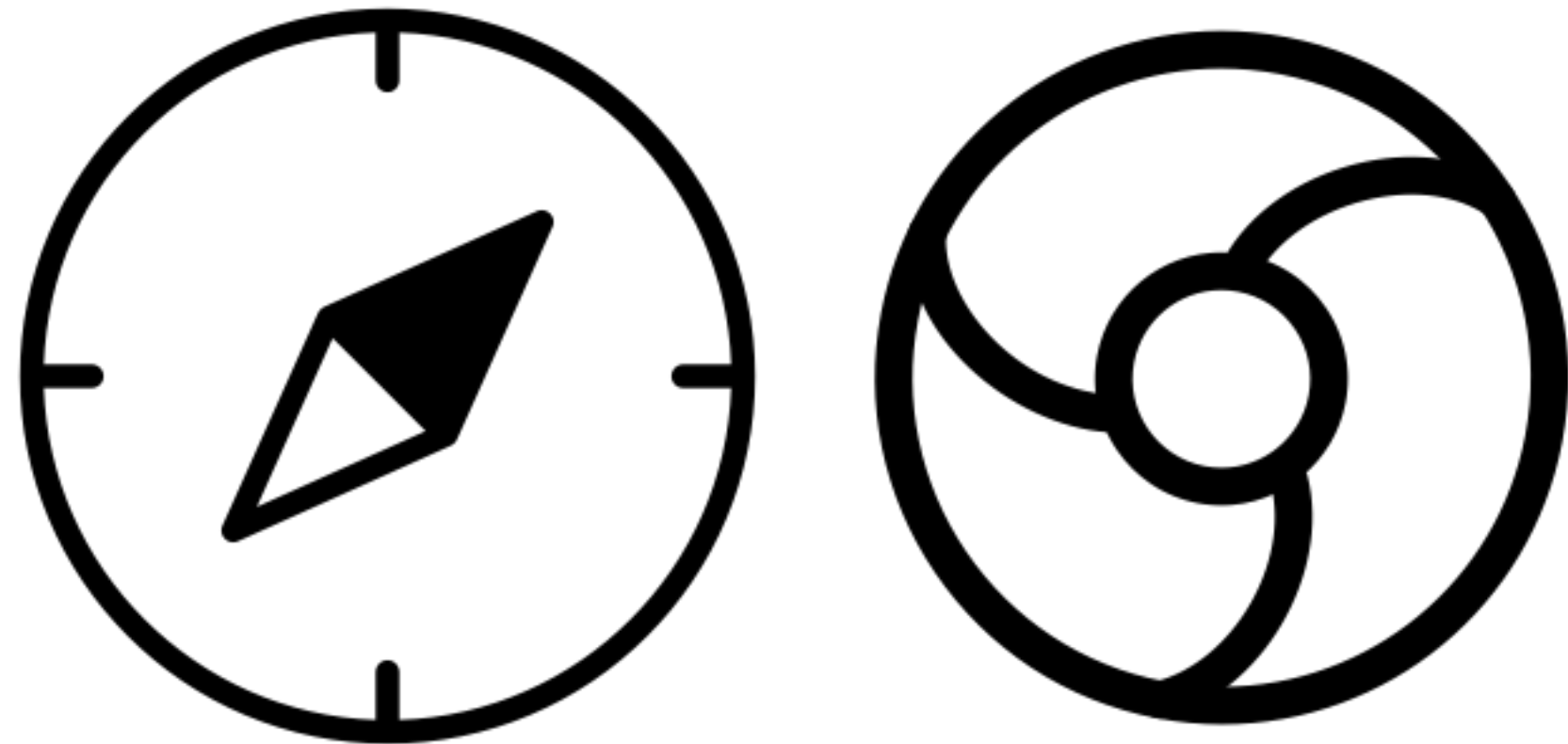
browsers



servers



http



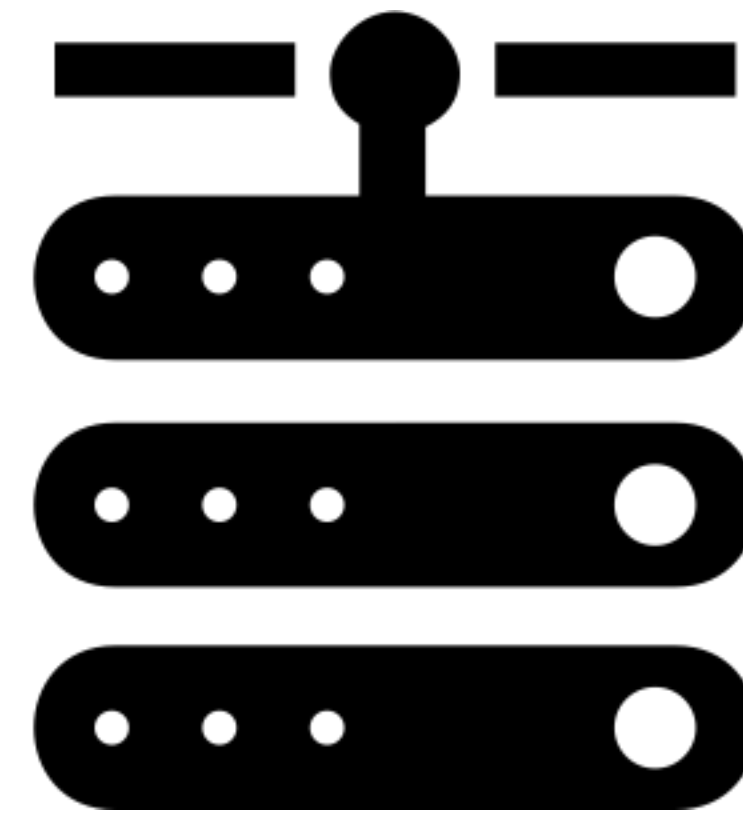
front-end

back-end

browsers



servers



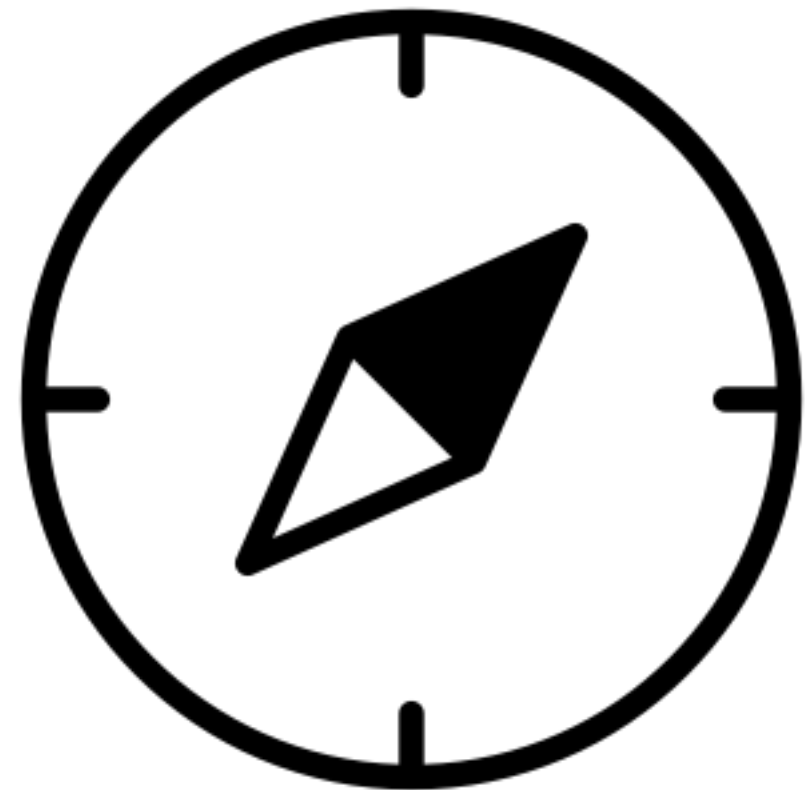
db



http



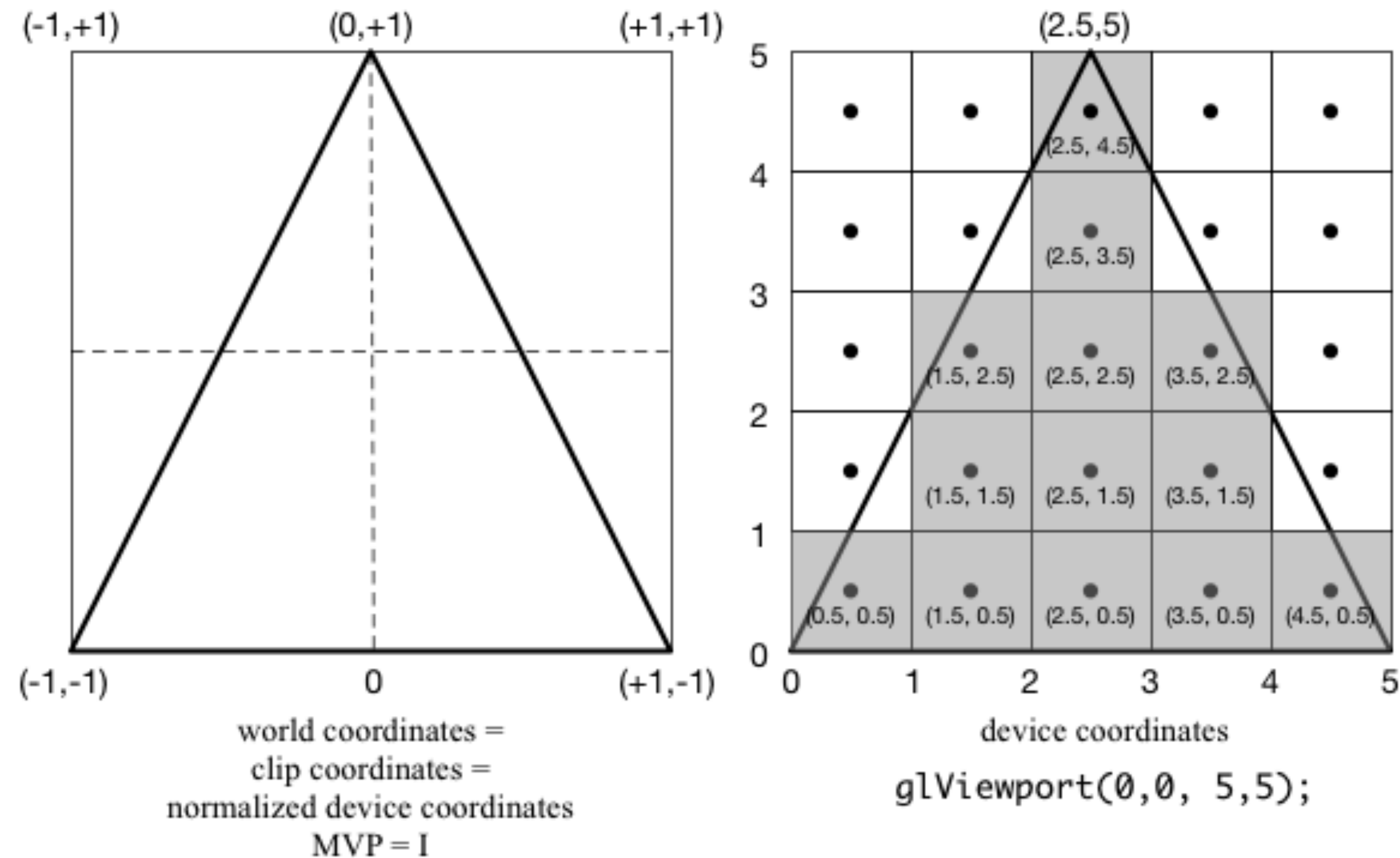
local
(net)



front-end

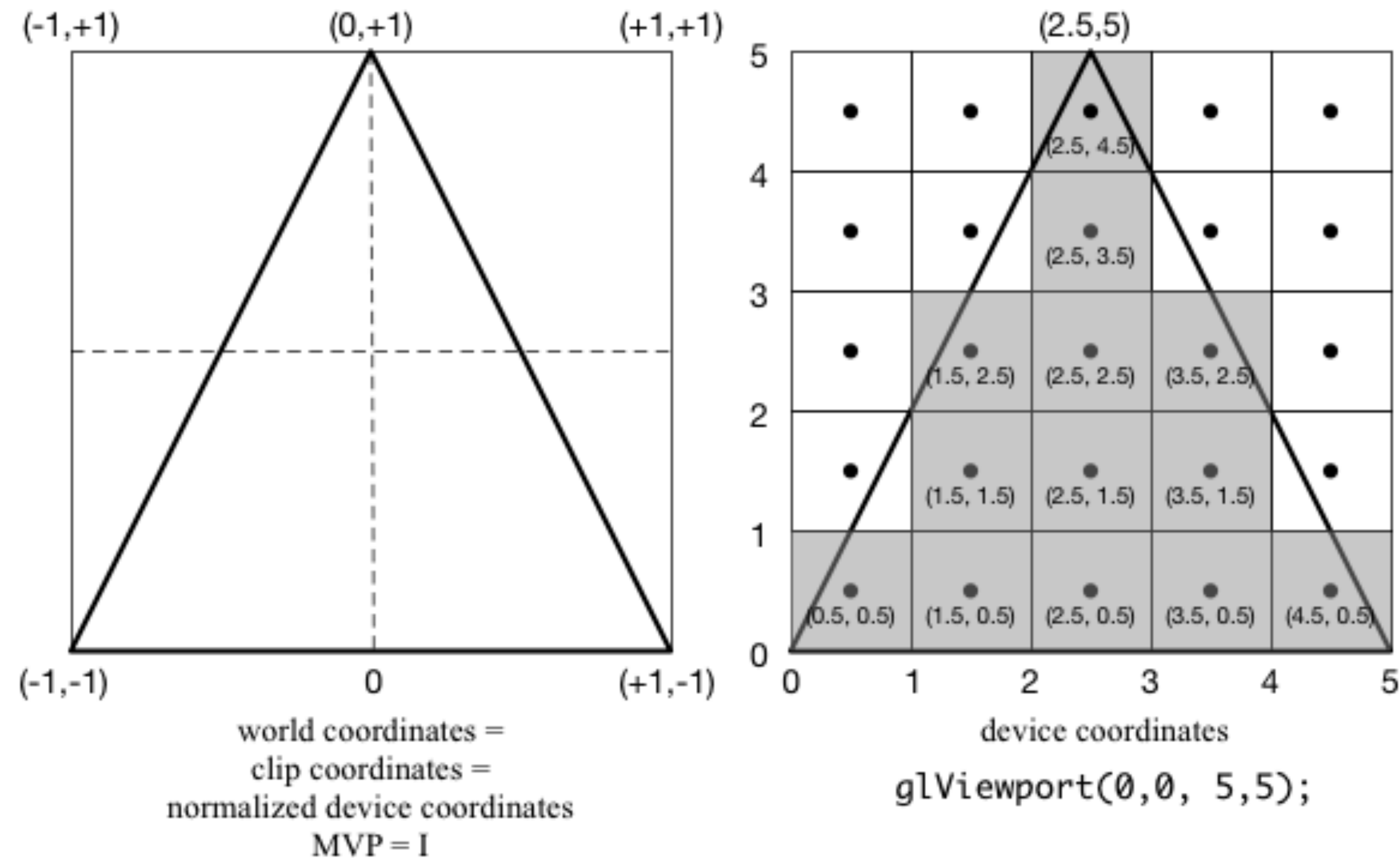
back-end

canvas vs. html



control pixels ->
build anything

canvas vs. html



control pixels ->
build anything



html ->
describe & render

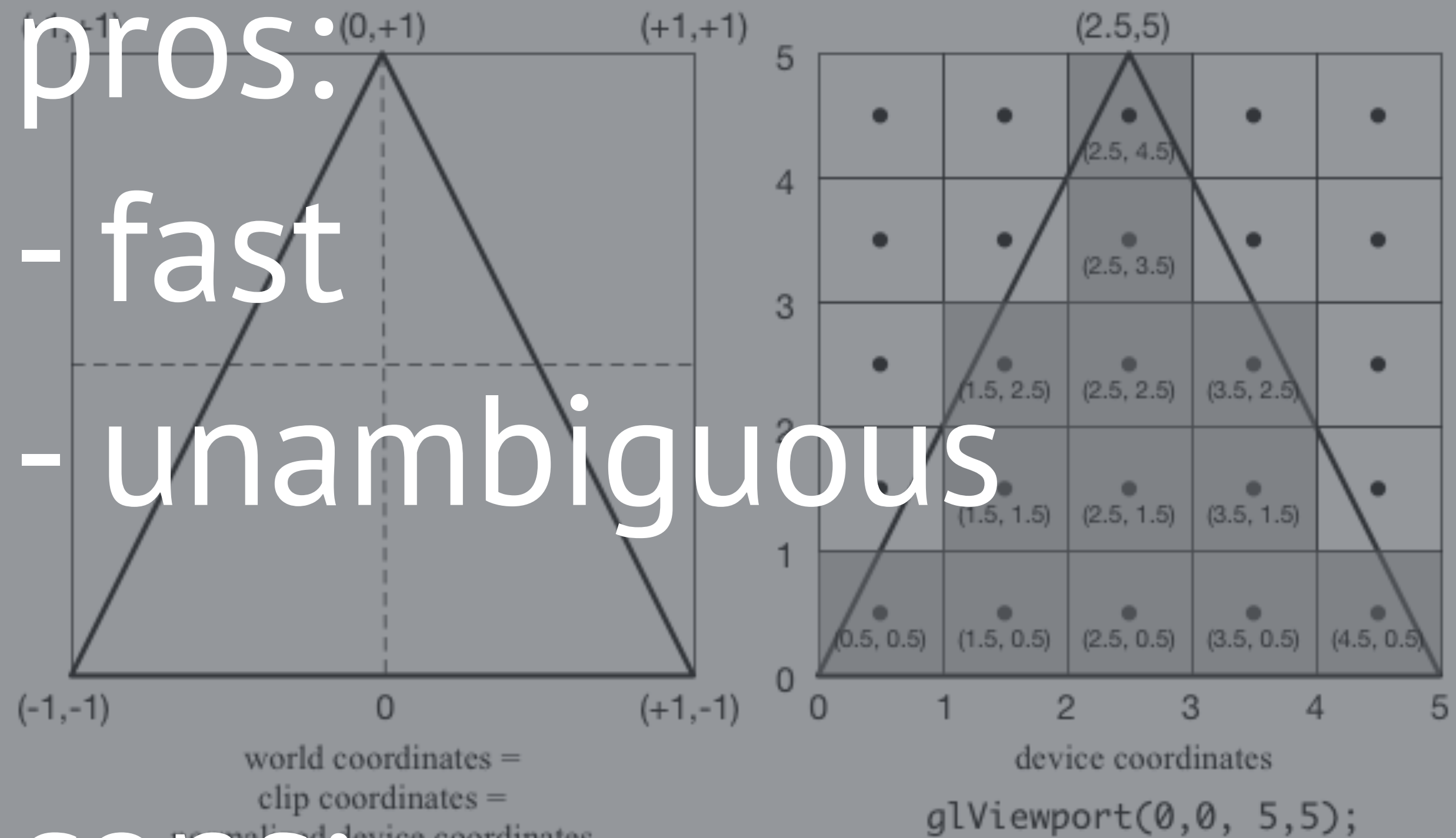
canvas vs. html

pros:

- fast
- unambiguous

cons:

control pixels ->
- unreadable
build anything



pros:


- readable
(human and machine)
- flexible

cons:

html ->
- slow(ish)
describe & render


html -> tags + content

<p>When the snows fall and the white winds blow,
the lone wolf dies but the pack survives.</p>



content

A vertical line with circular endpoints at the top and bottom, connecting the opening and closing tags of the paragraph above to the word 'content' below.




tag


A vertical line with circular endpoints at the top and bottom, connecting the opening and closing tags of the paragraph above to the word 'tag' below.

html -> tags + content


<p>When the snows fall and the white winds blow,
the lone wolf dies but the pack survives.</p>



content



tag

major  -> tags can

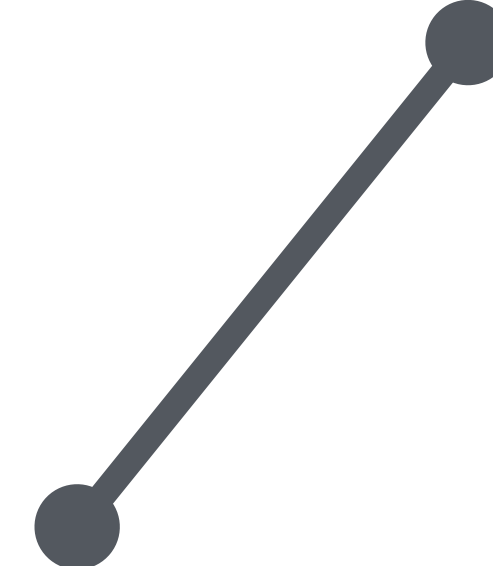
a) describe presentation, b) describe semantics, c) both

html -> tags + content


<p>When the snows fall and the white winds blow,
the lone wolf dies <i>but</i> the pack survives.</p>



presentation only:
italics



semantic + presentation:
paragraph, displayed as a
block

major  -> tags can

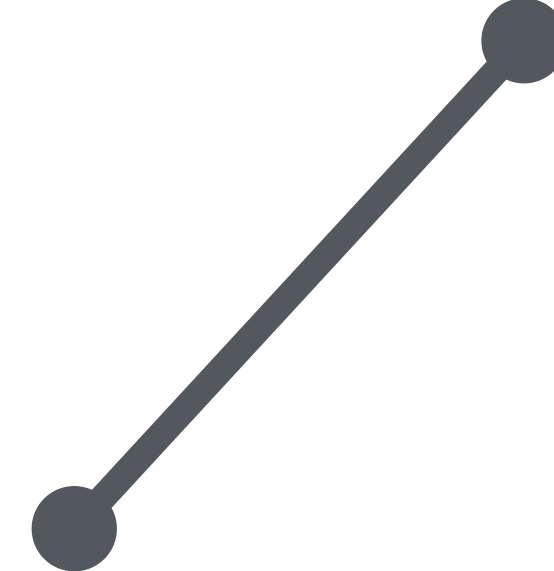
a) describe presentation, b) describe semantics, c) both

html -> tags + content


<div>When the snows fall and the white winds blow,
the lone wolf dies <i>but</i> the pack survives.</div>



presentation only:
italics



NOT semantic +
presentation: displayed as
a block

major  -> tags can

a) describe presentation, b) describe semantics, c) both

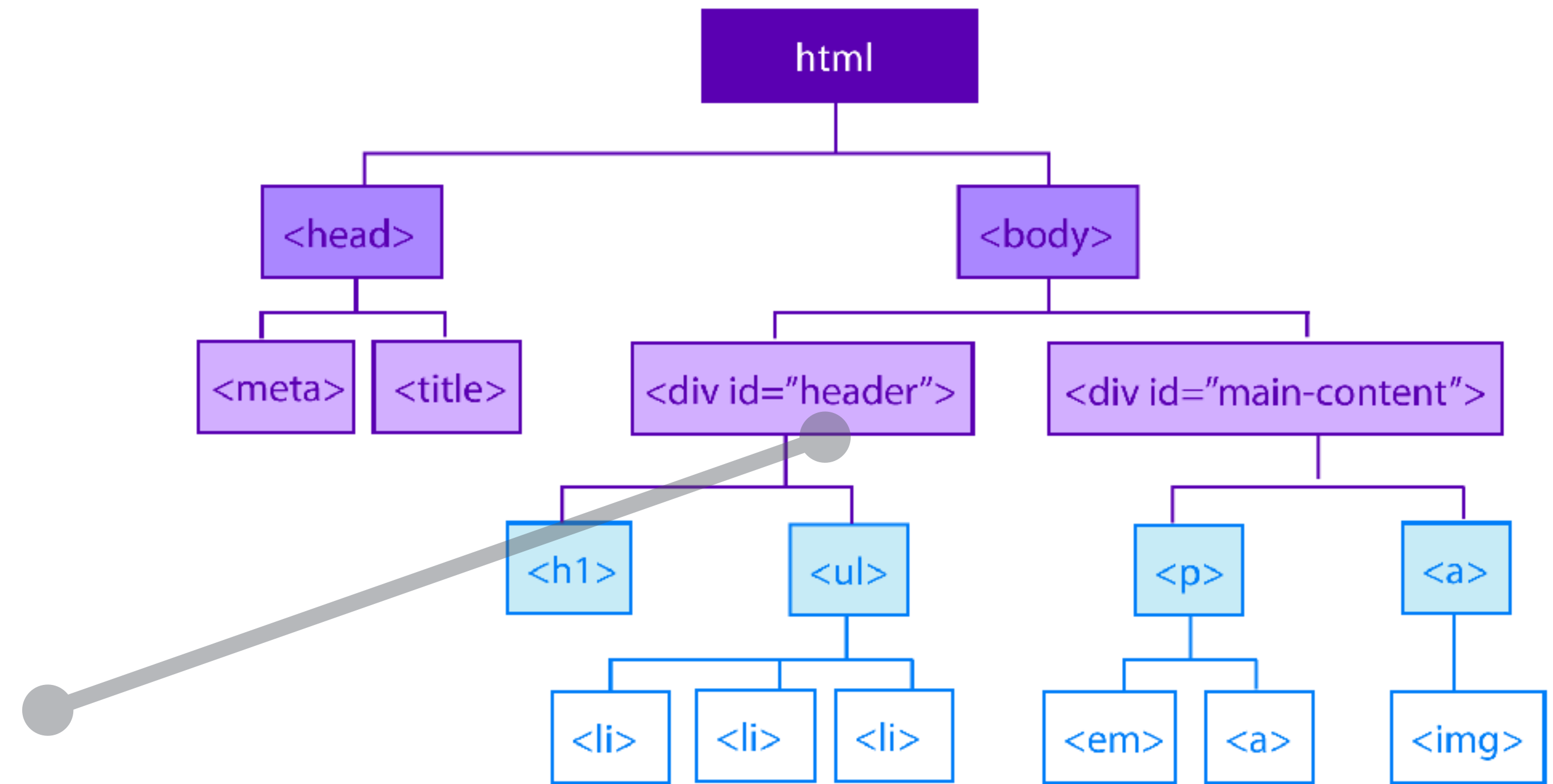
example:

let's make a page

html rules

- tree
- starts with html
- nesting allowed*
- must have start and end**
- tags can contain attributes

Simple Document Tree



other considerations

- whitespace:

usually not computed

- special characters:

{< > & “} -> {< > &l; "}

- train your eyes:

look for missing end-tags, quotes, etc.

(the same way you can spot a missing

parentheses or semicolon from a mile away)

```
<head>
  <title>Hello World
</head>
<body>
  <p class="computer">Hello world!</p>
</body>
```


typical (modern) structure

<head> <!-- imports, metadata -->

<style> <!-- page-specific style, or css import -->

<body> <!-- actual content, elements you will add to via js -->

<script> <!-- load js for dynamic content -->

when html is ambiguous...

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
```



make
developers consistent

Web technology support [\[edit \]](#)

Information about what web standards, and technologies the browsers support, except for JavaScript. External links lead to information about support in future versions of the browsers or extensions that provide such functionality.

| Browser | CSS2.1 ^[note 1] | Frames ^[note 2] | Nav
LINKs ^[note 3] | XSLT | XHTML
1.0 ^[note 4] | XHTML
1.1 ^[note 4] | MathML |
|---------------------------------------|--|--|---|----------------------|---|---|------------------------|
| Lunascape | Yes ^[87] | Yes | Depends ^[note 6] | Yes | Yes ^[87] | Yes ^[87] | Yes |
| Amaya | Yes | No | ? | No | Yes | Yes | Yes |
| Arora | Yes | Yes | No | Yes | Yes | Yes | Yes |
| Camino | Yes | Yes | No | Yes | Yes | Yes | Yes |
| Dooble | Yes | Yes | No | Yes | Yes | Yes | Yes |
| Flock | Yes | Yes | ? | Yes | Yes | Yes | Yes |
| Galeon | Yes | Yes | ? | Yes | Yes | Yes | Yes |
| Google
Chrome | Yes | Yes | No | Yes | Yes | Yes | No ^[8] |
| Internet
Explorer | Yes ^[note 7] | Yes | No ^[note 8] | Yes | Yes | Yes | No ^[note 8] |
| K-Meleon | Yes | Yes | ? | Yes | Yes | Yes | Yes |
| Konqueror | Yes | Yes | Yes | No | Yes | Yes | No |
| Microsoft Edge | Yes | Yes | No | Yes | Yes | Yes | No |

make
browsers consistent

when html is ambiguous...

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
```



like herding
cats.
make
developers consistent

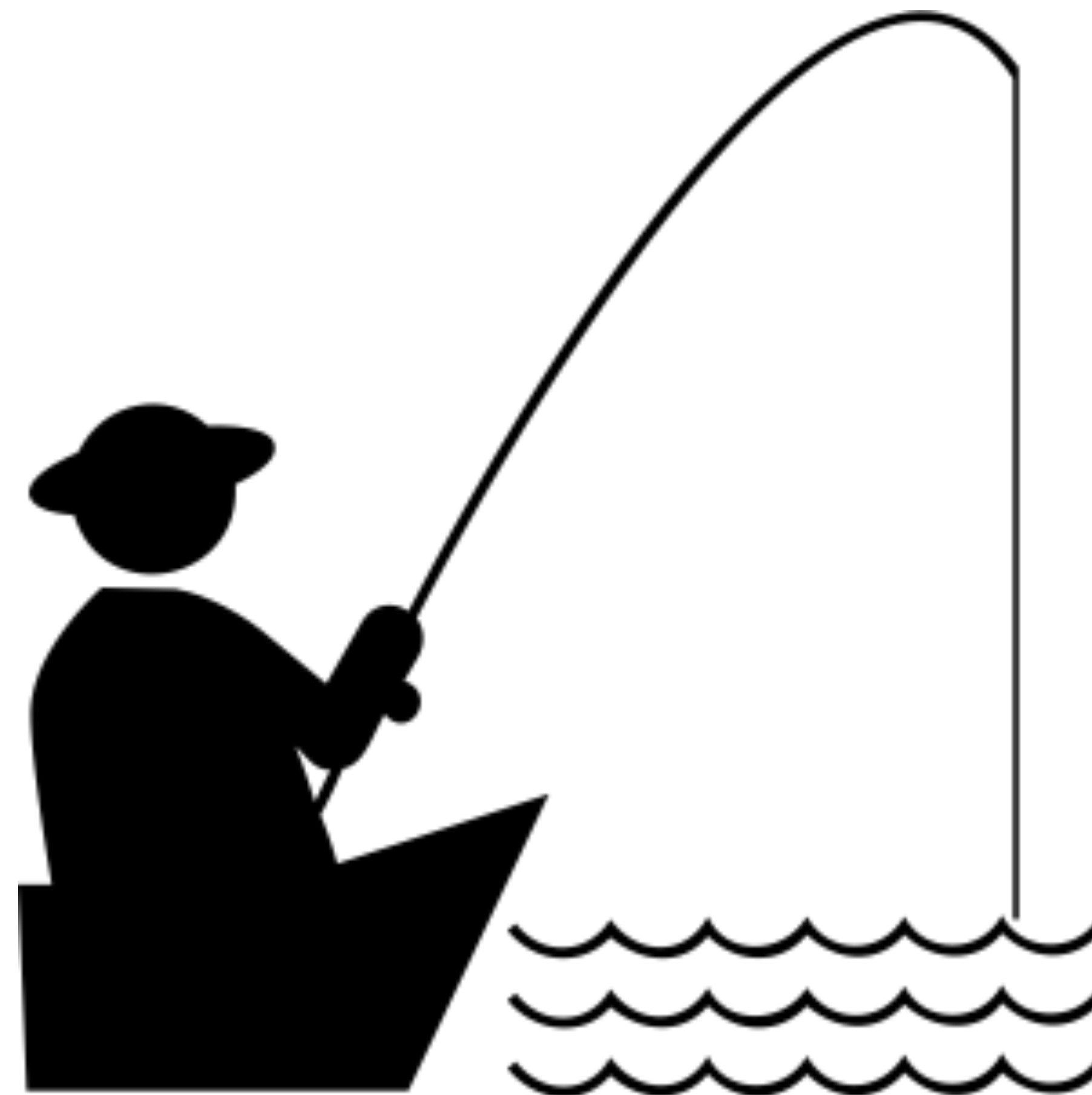
can't keep up with
new tech
make
browsers consistent

Web technology support [\[edit \]](#)

Information about what web standards, and technologies the browsers support, except for JavaScript. External links lead to information about support in future versions of the browsers or extensions that provide such functionality.

| Browser | CS52.1 ^[note 1] | Frames ^[note 2] | Nav
LINKs ^[note 3] | XSLT | XHTML
1.0 ^[note 4] | XHTML
1.1 ^[note 4] | MathML |
|----------------------|----------------------------|----------------------------|----------------------------------|------|----------------------------------|----------------------------------|------------------------|
| Lunascape | Yes ^[87] | Yes | Depends ^[note 6] | Yes | Yes ^[87] | Yes ^[87] | Yes |
| Amaya | Yes | No | ? | No | Yes | Yes | Yes |
| Arora | Yes | Yes | No | Yes | Yes | Yes | Yes |
| Camino | Yes | Yes | No | Yes | Yes | Yes | Yes |
| Dooble | Yes | Yes | No | Yes | Yes | Yes | Yes |
| Envy | Yes | Yes | ? | Yes | Yes | Yes | Yes |
| Firefox | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Google
Chrome | Yes | Yes | No | Yes | Yes | Yes | No ^[9] |
| Internet
Explorer | Yes ^[note 7] | Yes | No ^[note 8] | Yes | Yes | Yes | No ^[note 9] |
| K-Meleon | Yes | Yes | ? | Yes | Yes | Yes | Yes |
| Konqueror | Yes | Yes | Yes | No | Yes | Yes | No |
| Microsoft Edge | Yes | Yes | No | Yes | Yes | Yes | No |

HTML5



(learning to fish)

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It was finalized, and published, on 28 October 2014 by the World Wide Web Consortium (W3C). This is the fifth revision of the HTML standard since the inception of the World Wide Web.

[HTML5 - Wikipedia, the free encyclopedia](https://en.wikipedia.org/wiki/HTML5)

<https://en.wikipedia.org/wiki/HTML5> Wikipedia ▾

[Block en.wikipedia.org](#)



More about HTML5

Feedback

<html>

<head>

<bbody>

<p>

<h1...6>

<table>

<list>

CSS

why use css?

html -> semantics and presentation

why use css?

html -> semantics and presentation

it makes sense to separate semantics and presentation:

- mobile browsers
- accessibility
- handoff to team (designers, devs)
- adaptability for user preferences

style sheet levels

`<div style="color:red">This txt is red</div>`

in-line

style sheet levels

`<div style="color:red">This txt is red</div>`

in-line

`<style>div { color: red; }</style>`

in-page

`<div>This txt is red</div>`

`<div>This txt is red, too</div>`

style sheet levels

`<div style="color:red">This txt is red</div>`

in-line

`<style>div { color: red; }</style>`

in-page

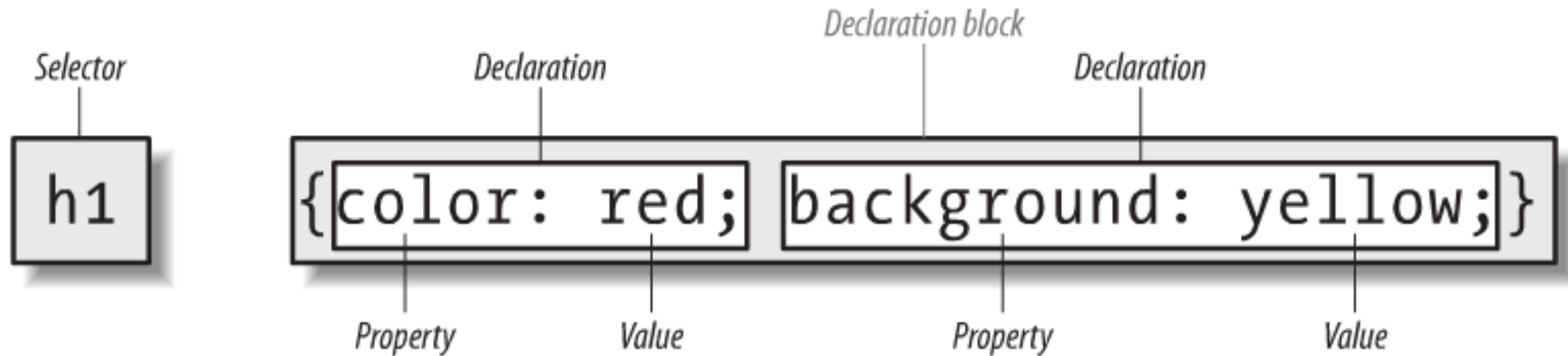
`<div>This txt is red</div>`

`<div>This txt is red, too</div>`

`<link rel="stylesheet" href="style.css">`

external

css rule structure



selector types

p { color: blue; }

single

p, h2 {color: blue;}

grouped

* {color: blue;}

universal

these are the same

```
/* group 1 */
h1 {color: silver; background: white;}
h2 {color: silver; background: gray;}
h3 {color: white; background: gray;}
h4 {color: silver; background: white;}
b {color: gray; background: white;}

/* group 2 */
h1, h2, h4 {color: silver;}
h2, h3 {background: gray;}
h1, h4, b {background: white;}
h3 {color: white;}
b {color: gray;}

/* group 3 */
h1, h4 {color: silver; background: white;}
h2 {color: silver;}
h3 {color: white;}
h2, h3 {background: gray;}
b {color: gray; background: white;}
```

selector depth (descendant)

```
div span img { background-color: blue;}
```

an image, inside an ordered list, inside a div

classes (.)

Used for types:

examples: .status; .warning; .song

```
.warning {  
  color: red;  
  font-size: 28;  
}
```

can be combined! `<p class="warning status"></p>`

ids (#)

Used for types, but ONLY ONE:
examples: #headlogo; #login

```
#headlogo {  
  width: 100%;  
  height: 200px;  
}
```

```

```



example: doing useful things in css...

which rule wins?

1. SPECIFICITY
2. INHERITANCE
3. CASCADE

which rule wins?

1. SPECIFICITY

2. INHERITANCE

3. CASCADE

1. SPECIFICITY

2. INHERITANCE

3. CASCADE

1. SPECIFICITY

2. INHERITANCE

3. CASCADE

FIND OUT NEXT TIME...

1. SPECIFICITY

2. INHERITANCE

3. CASCADE

2. INHERITANCE

3. CASCADE

1. SPECIFICITY

2. INHERITANCE

3. CASCADE

1. SPECIFICITY

2. INHERITANCE

3. CASCADE

1. SPECIFICITY

2. INHERITANCE

3. CASCADE