JÖNKÖPING UNIVERSITY

School of Engineering

CSS

Peter Larsson-Green
Jönköping University
Autumn 2018

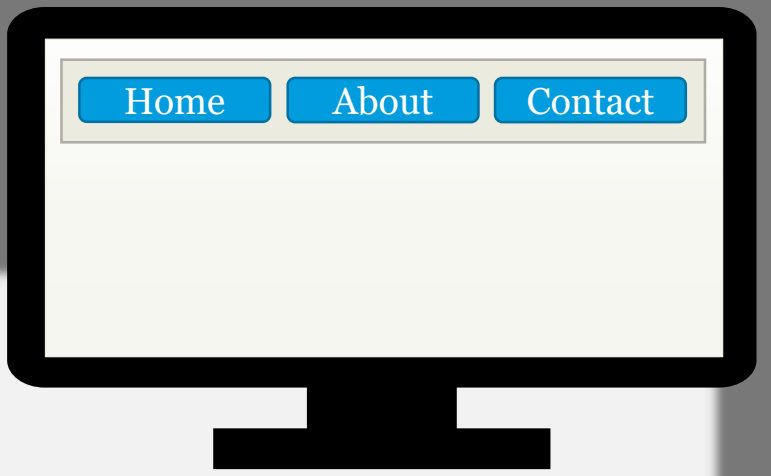
CSS

HTML: Mark what type of data text represents.

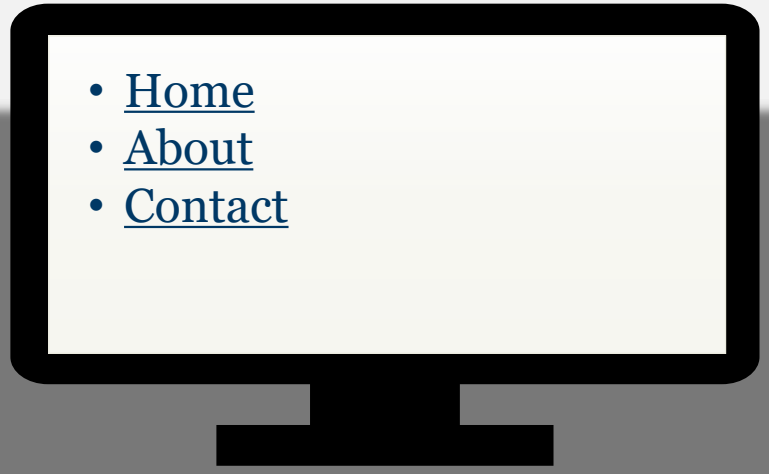
- Web browsers render the webpage.

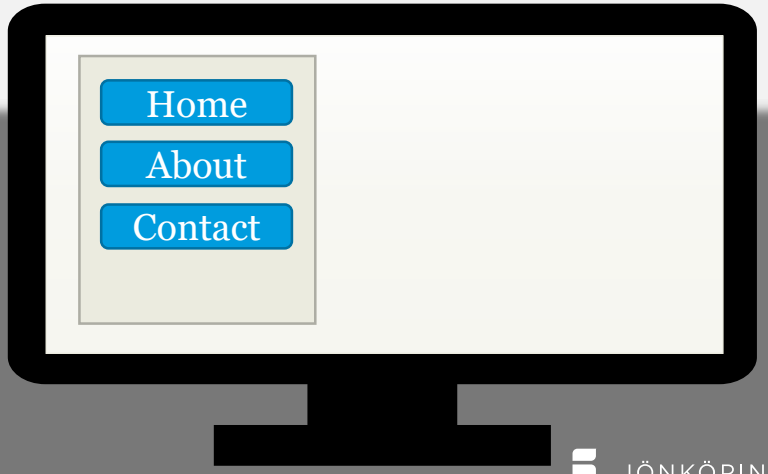
CSS: Tell the web browsers how to render the data.

CSS EXAMPLE



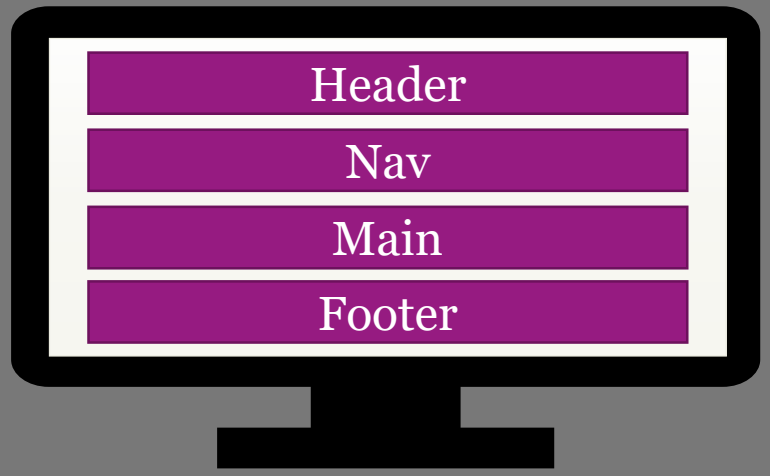

```
<nav>
  <ul>
    <li><a href="home.html">Home</a></li>
    <li><a href="aboute.html">About</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</nav>
```

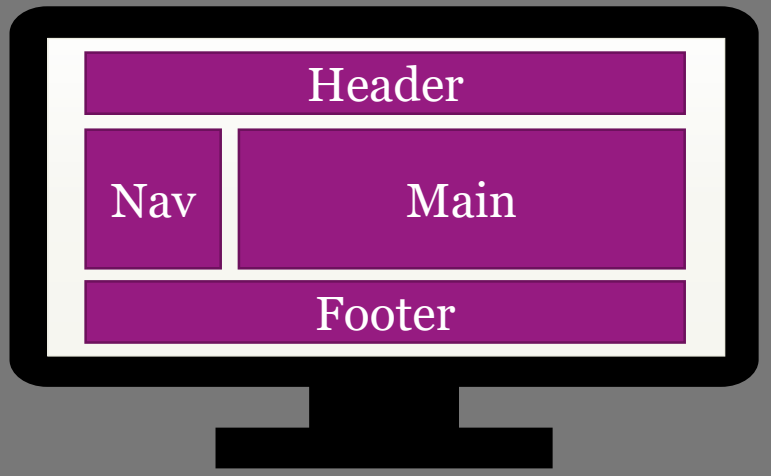




CSS EXAMPLE

```
<header>Header</header>
<nav>Nav</nav>
<main>Main</main>
<footer>Footer</footer>
```





CSS LEVELS

CSS 2.1 was a *candidate recommendation* 2004.

CSS 3 consists of modules.

- Some have finished specifications.
- Some have almost finished specifications.
- Some are still early drafts.

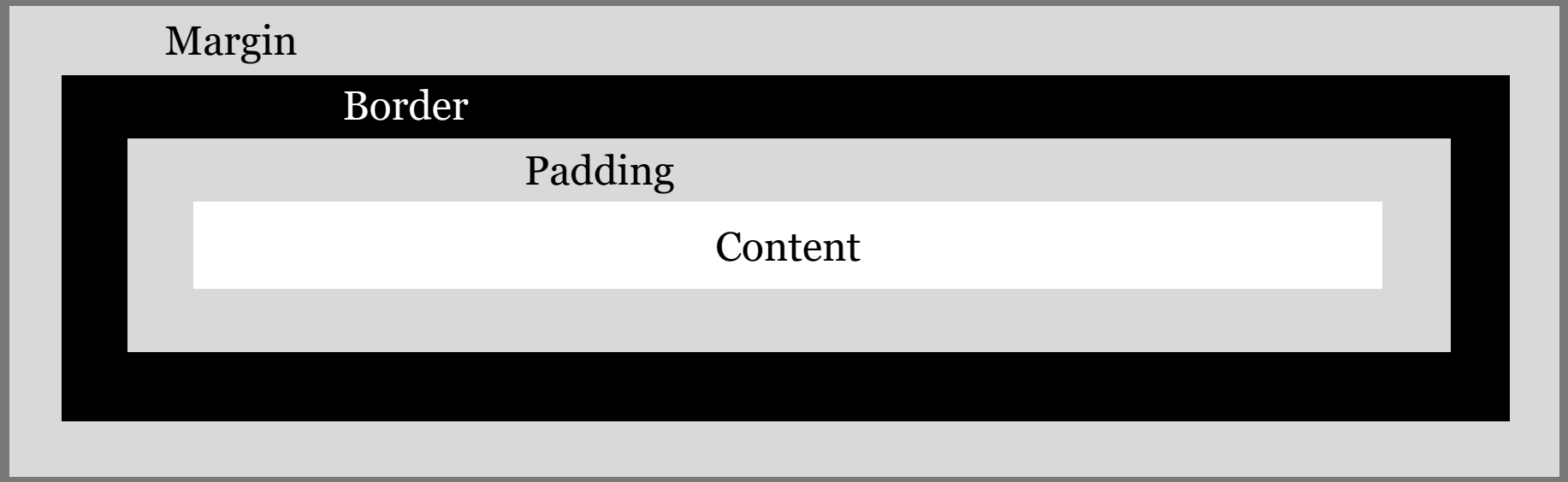
CSS 4 continues with modules.

https://caniuse.com

THE BOX MODEL

Explains how web browsers render elements.

- All elements are rendered as boxes:



WHERE TO WRITE CSS CODE

1. In the global `style` attribute:

```
<p style="CSS-CODE">Some text</p>
```

   - Can't re-use our CSS code on other elements ☹

2. In the `<style>` element:

```
<style>CSS-CODE</style>
```

   - Need to specify which elements that should be affected (selectors).
   - Can't re-use our CSS code in other HTML files ☹

WHERE TO WRITE CSS CODE

3. In a separate `.css` file:

```
<link rel="stylesheet" href="the-css-file.css">
```

```
CSS-CODE
```

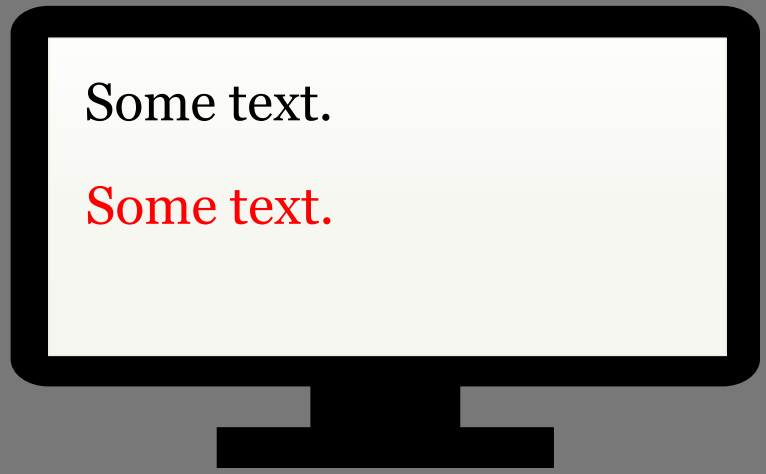
- Need to specify which elements that should be affected (selectors).
- Can use the same CSS code in multiple HTML files ☺
- CSS files can be cached ☺

CSS SYNTAX

Declaration:

```
property-name: value;
```

```
<p>Some text.</p>
<p style="color: red">Some text.</p>
```



CSS SYNTAX

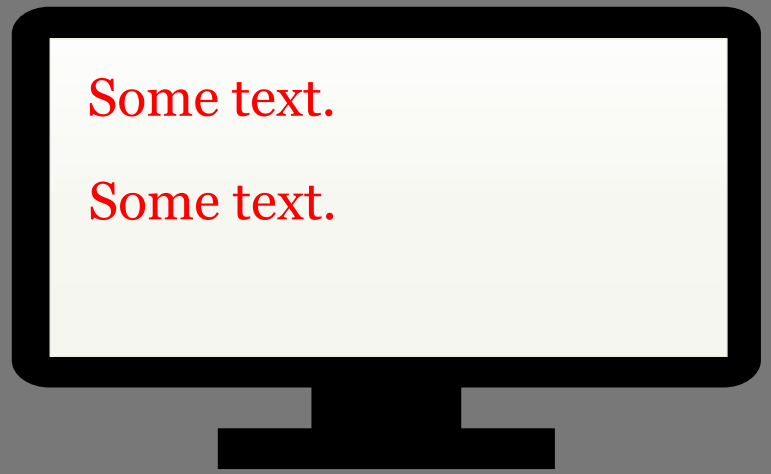
Declaration:

```
property-name: value;
```

Rule:

```
selector{
  declarations
}
```

```
<style>
p{
  color: red
}
</style>
<p>Some text.</p>
<p>Some text.</p>
```



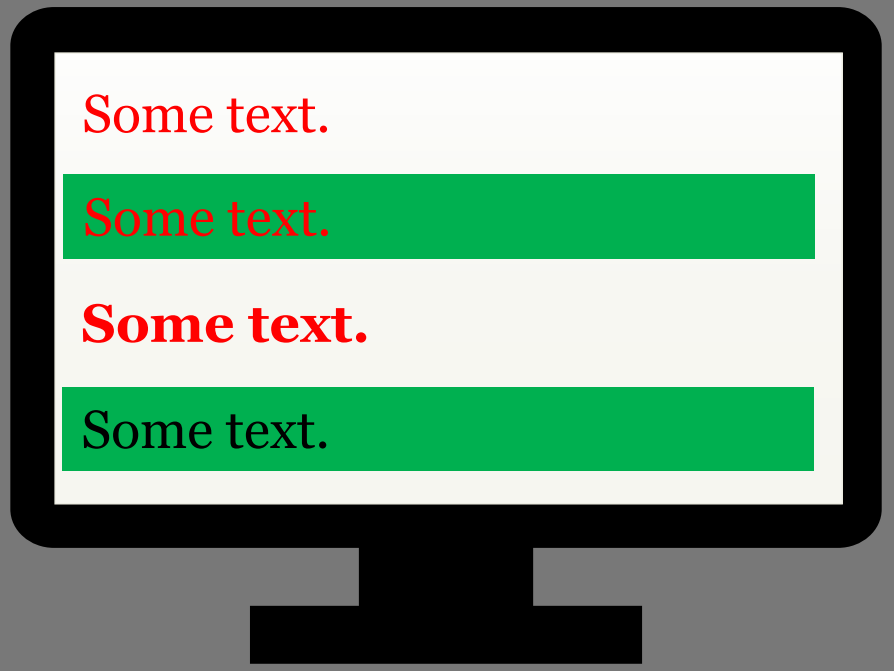
CSS SELECTORS

`tagname`	The elements with the tag `tagname`.
`#the-id`	The element with the attribute: `id="the-id"`
`.a-class-name`	The elements with the attribute: `class="a-class-name"`
`*`	All elements.

EXAMPLE

```
<style>
  p{ color: red }
  #cool{ font-weight: bold }
  .happy{ background-color: lime }
</style>

<p>Some text.</p>
<p class="happy">Some text.</p>
<p id="cool">Some text.</p>
<div class="happy">Some text.</div>
```



RELATIONAL SELECTORS

```
selectorA selectorB
```

The elements mathed by `selectorB` that are inside an element matched by `selectorA`.

```
selectorA > selectorB
```

The elements matched by `selectorB` that are direct children to an element matched by `selectorA`.

```
selectorA + selectorB
```

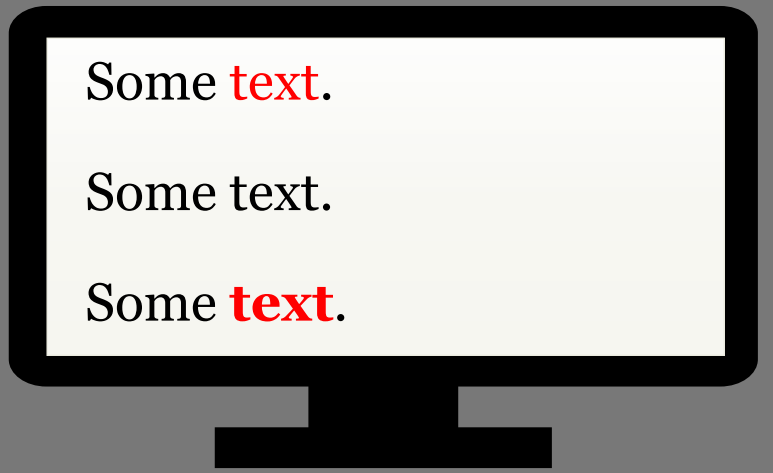
The elements matched by `selectorB` that comes directly after an element matched by `selectorA`.

And more!

EXAMPLE

```
<style>
  p span{ color: red }
</style>

<p>Some <span>text</span>.</p>
<span>Some text.</span>
<p>
  Some <strong><span>text</span></strong>.
</p>
```



EXAMPLE

```
<style>
  p > span{ color: red }
</style>

<p>Some <span>text</span>.</p>
<span>Some text.</span>
<p>
  Some <strong><span>text</span></strong>.
</p>
```

MULTIPLE SELECTORS

```
selectorA, selectorB
```

The elements mathed by `selectorA` or `selectorB`.

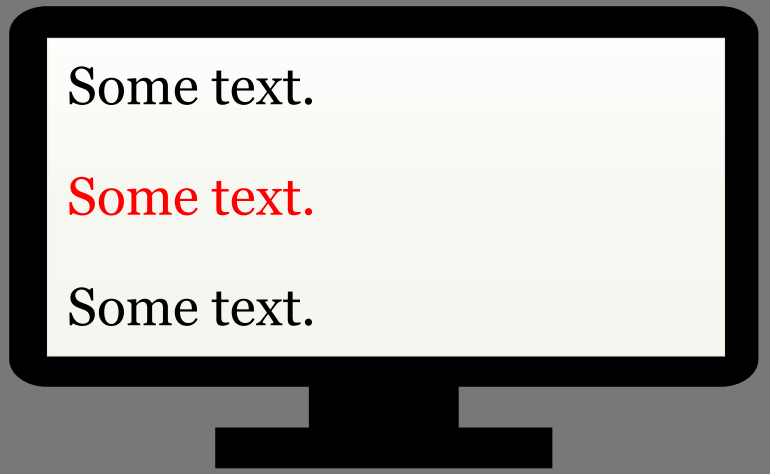
```
selectorAselectorB
```

The elements matched by `selectorA` and `selectorB`.

EXAMPLE

```
<style>
  p.happy{
    color: red
  }
</style>

<p>Some text.</p>
<p class="happy">Some text.</p>
<span class="happy">Some text</span>
```



SELECTORS WITH PSEUDO-CLASSES

`theSelector:first-child`	The elements matched by `theSelector` when they are the first child in its parent.
`theSelector:focus`	The elements matched by `theSelector` when they has focus.
`theSelector:hover`	The elements matched by `theSelector` when the mouse hovers over them.
`theSelector:visited`	The links matched by `theSelector` when they have been visited.

And more!

SELECTORS WITH ATTRIBUTES

```
theSelector[attr]
```

The elements matched by `theSelector` and have the attribute `attr`.

```
theSelector[attr=value]
```

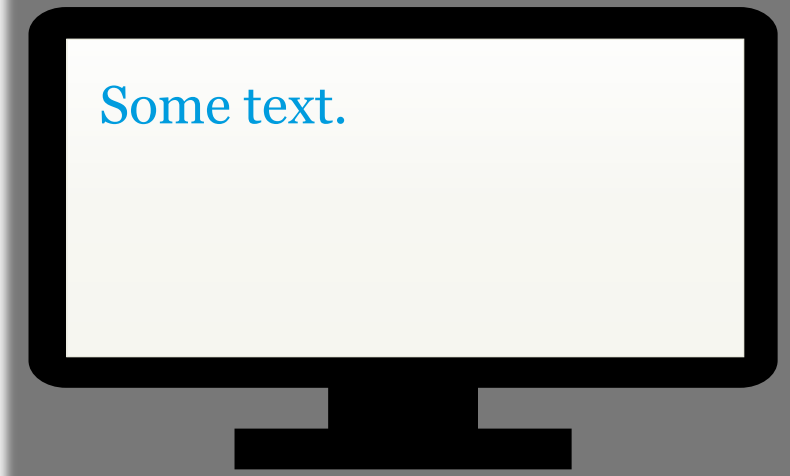
The elements matched by `theSelector` and have `attr="value"`.

And more!

CONFLICTING RULES

```
<style>
  p{ color: red }
  #cool{ color: blue }
  .happy{ color: yellow }
</style>

<p id="cool" class="happy">Some text.</p>
```



SELECTOR SPECIFICITY

https://www.w3.org/TR/css3-selectors/#specificity

1. The style attribute.
2. Rule with most id selectors.
3. Rule with most class selectors.
4. Rule with most tag name selectors.

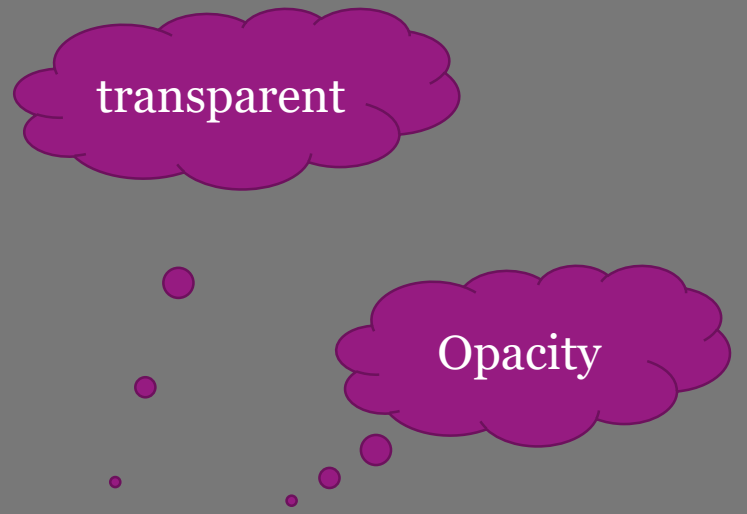
SPECIFICITY EXAMPLE

```
<style>
  #hi .hello p{ color: red }
  #hi p { color: blue }
  #para { color: yellow }
</style>

<main id="hi">
  <div class="hello">
    <p id="para">Some text</p>
  </div>
</main>
```

CSS COLOR VALUES

- The name of the color.
  - https://www.w3.org/TR/css-color-3/#html4
  - https://www.w3.org/TR/css-color-3/#svg-color
- `transparent`
- `rgb(R, G, B)` 0 <= R, G, B <= 255
- `rgba(R, G, B, a)` 0 <= R, G, B <= 255, 0 <= a <= 1
- `#RRGGBB` 00 <= RR, GG, BB <= FF



CSS UNITS

https://www.w3.org/Style/Examples/007/units.en.html

- Absolute:
  - cm, mm, in, px, pt…
- Relative:
  - % - percentage of parent
  - em - relative to parent font size
  - vw - % of view width
  - vh - % of view height
  - vmin - % of the smallest of the view width and view height
  - vmax - % of the biggest of the view width and view height

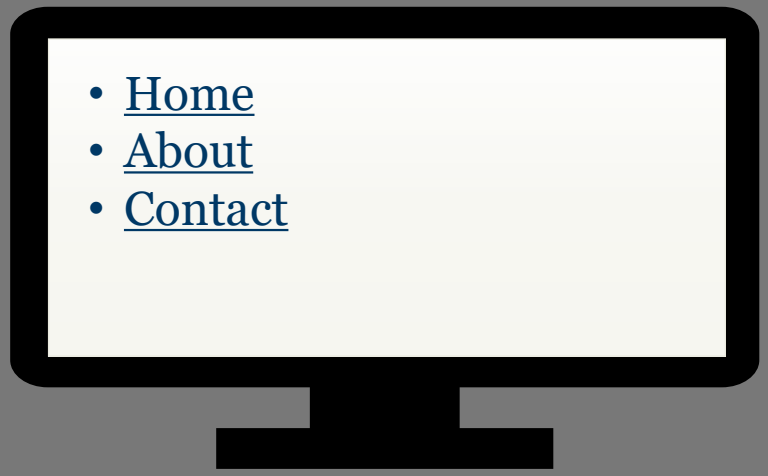
CSS PROPERTIES

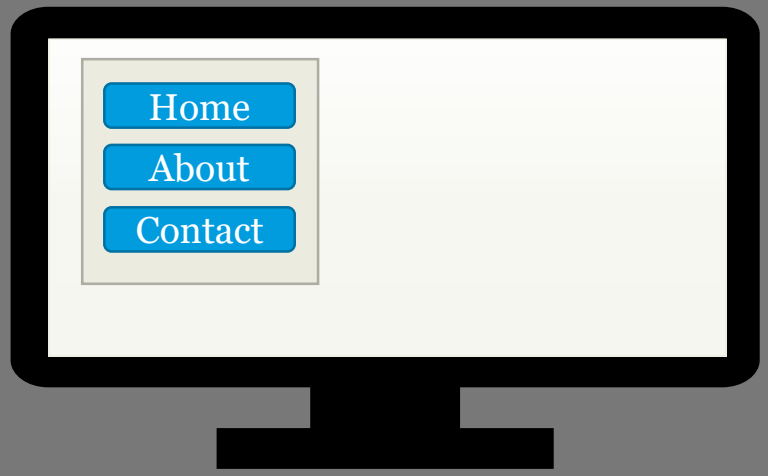
List of most properties: https://www.w3.org/Style/CSS/all-properties.en.html

Practical demonstration…

PRACTICAL EXAMPLE

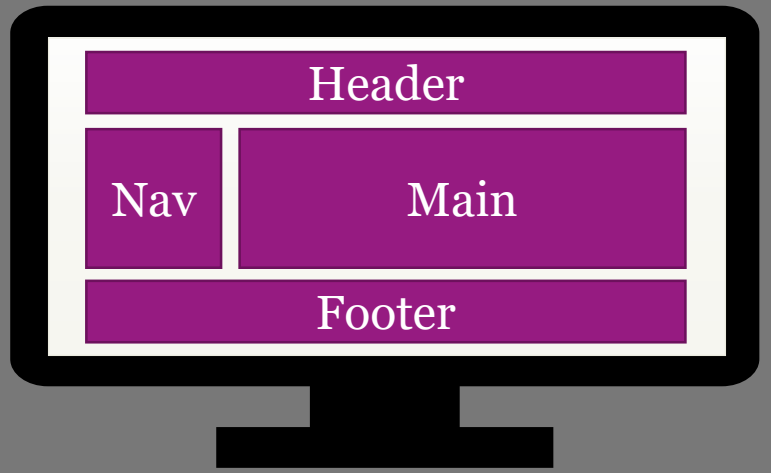
That menu in the beginning…





LAYOUT EXAMPLE

Learn how to create layouts: http://learnlayout.com



MEDIA QUERIES

```
<link rel="stylesheet" href="file.css" media="MEDIA-QUERY">
```

```
<style>
  @media MEDIA-QUERY {
    /* Ordinary CSS code (e.g. rules). */
  }
</style>
```

MEDIA QUERY EXAMPLE

```
<style>
  @media screen and (max-width: 300px){
    /* CSS code for small screens. */
  }
  @media screen and (min-width: 301px){
    /* CSS code for big screens. */
  }
</style>
```