Lecture 04: Positions, Fonts, Responsive Web Design

ITP 303 Full-Stack Web Development

Key Takeaways Today

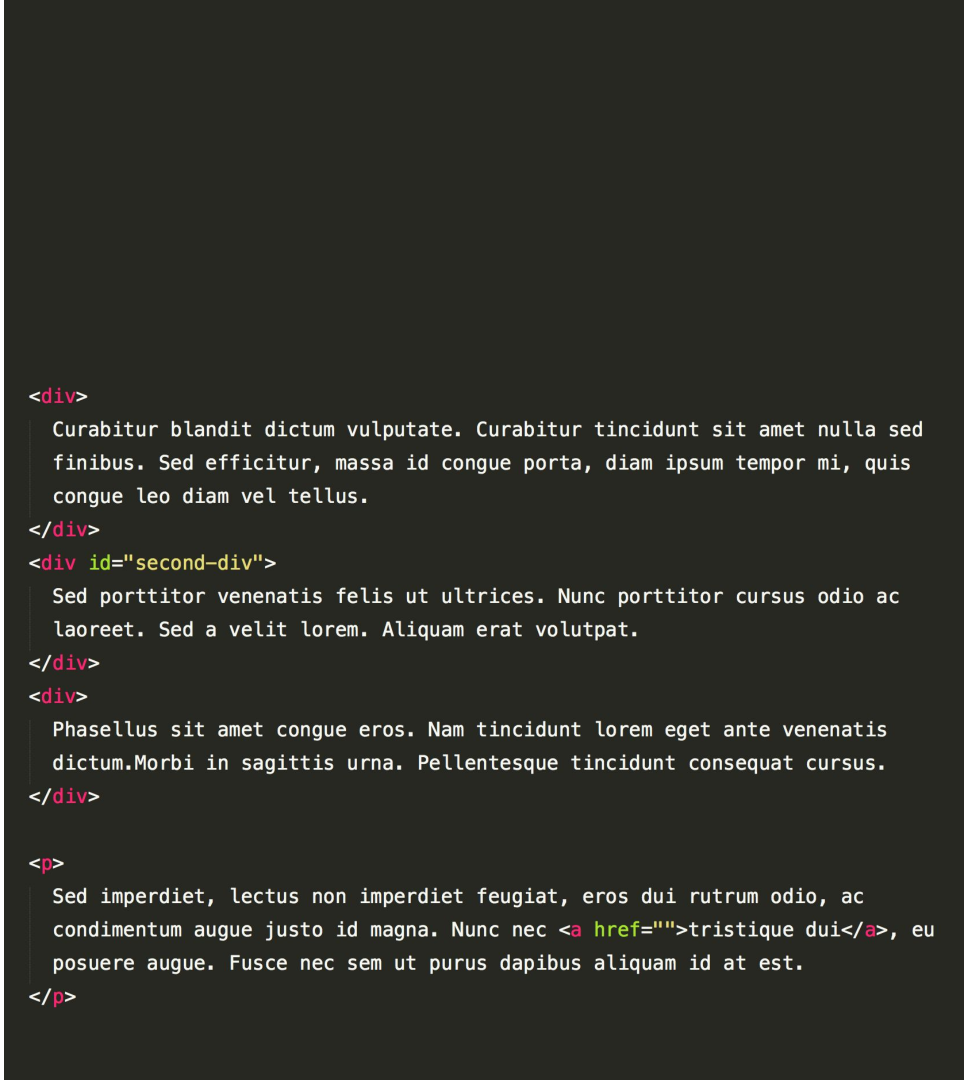
1. Inline vs block elements
2. CSS Positions
3. Hiding/Showing elements
4. CSS `box-sizing`
5. Responsive web design techniques
    a. Using media queries
    b. Using percentages
    c. Using min-width and max-width

Inline vs block elements

CSS `display`

Specifies how elements are displayed.

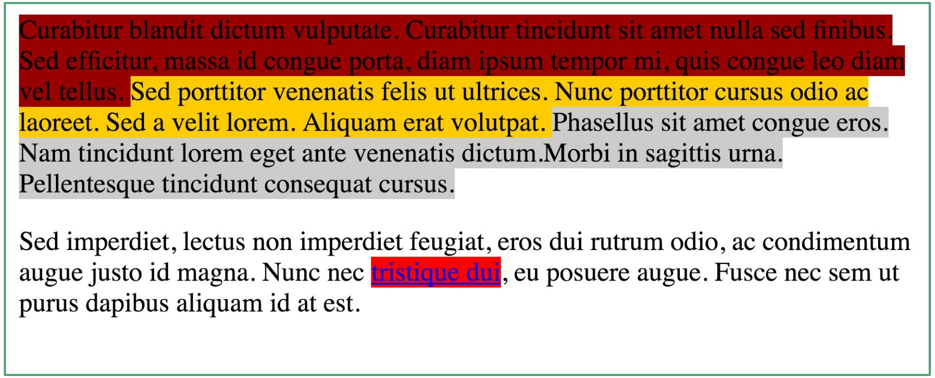
`block`	Displayed as a block. Default value for `<div>` and `<p>`.
`inline`	Displayed as an inline element. Default value for `<a>`.
`none`	Not displayed. Does not take up any space on the page.



```html
<div>
  Curabitur blandit dictum vulputate. Curabitur tincidunt sit amet nulla sed
  finibus. Sed efficitur, massa id congue porta, diam ipsum tempor mi, quis
  congue leo diam vel tellus.
</div>
<div id="second-div">
  Sed porttitor venenatis felis ut ultrices. Nunc porttitor cursus odio ac
  laoreet. Sed a velit lorem. Aliquam erat volutpat.
</div>
<div>
  Phasellus sit amet congue eros. Nam tincidunt lorem eget ante venenatis
  dictum.Morbi in sagittis urna. Pellentesque tincidunt consequat cursus.
</div>

<p>
  Sed imperdiet, lectus non imperdiet feugiat, eros dui rutrum odio, ac
  condimentum augue justo id magna. Nunc nec <a href="">tristique dui</a>, eu
  posuere augue. Fusce nec sem ut purus dapibus aliquam id at est.
</p>
```

CSS display

Result:



Curabitur blandit dictum vulputate. Curabitur tincidunt sit amet nulla sed finibus. Sed efficitur, massa id congue porta, diam ipsum tempor mi, quis congue leo diam vel tellus. Sed porttitor venenatis felis ut ultrices. Nunc porttitor cursus odio ac laoreet. Sed a velit lorem. Aliquam erat volutpat. Phasellus sit amet congue eros. Nam tincidunt lorem eget ante venenatis dictum.Morbi in sagittis urna. Pellentesque tincidunt consequat cursus.

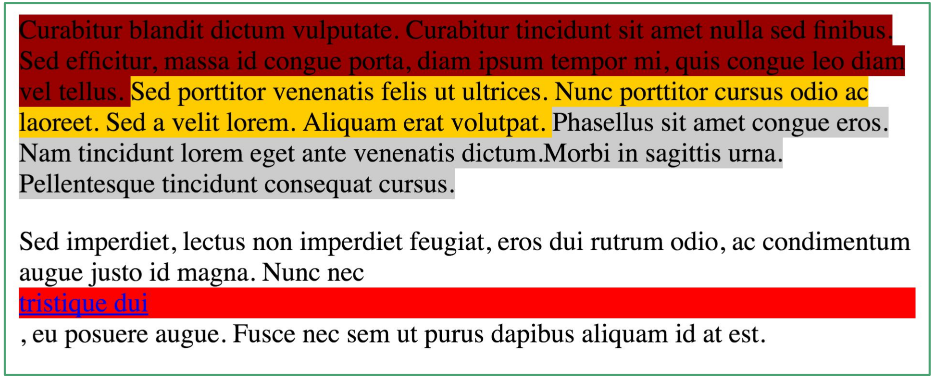
Sed imperdiet, lectus non imperdiet feugiat, eros dui rutrum odio, ac condimentum augue justo id magna. Nunc nec tristique dui, eu posuere augue. Fusce nec sem ut purus dapibus aliquam id at est.

```css
div {
  display: inline;
}
```

```html
<div>
  Curabitur blandit dictum vulputate. Curabitur tincidunt sit amet nulla sed
  finibus. Sed efficitur, massa id congue porta, diam ipsum tempor mi, quis
  congue leo diam vel tellus.
</div>
<div id="second-div">
  Sed porttitor venenatis felis ut ultrices. Nunc porttitor cursus odio ac
  laoreet. Sed a velit lorem. Aliquam erat volutpat.
</div>
<div>
  Phasellus sit amet congue eros. Nam tincidunt lorem eget ante venenatis
  dictum.Morbi in sagittis urna. Pellentesque tincidunt consequat cursus.
</div>

<p>
  Sed imperdiet, lectus non imperdiet feugiat, eros dui rutrum odio, ac
  condimentum augue justo id magna. Nunc nec <a href="">tristique dui</a>, eu
  posuere augue. Fusce nec sem ut purus dapibus aliquam id at est.
</p>
```

CSS display

Result:

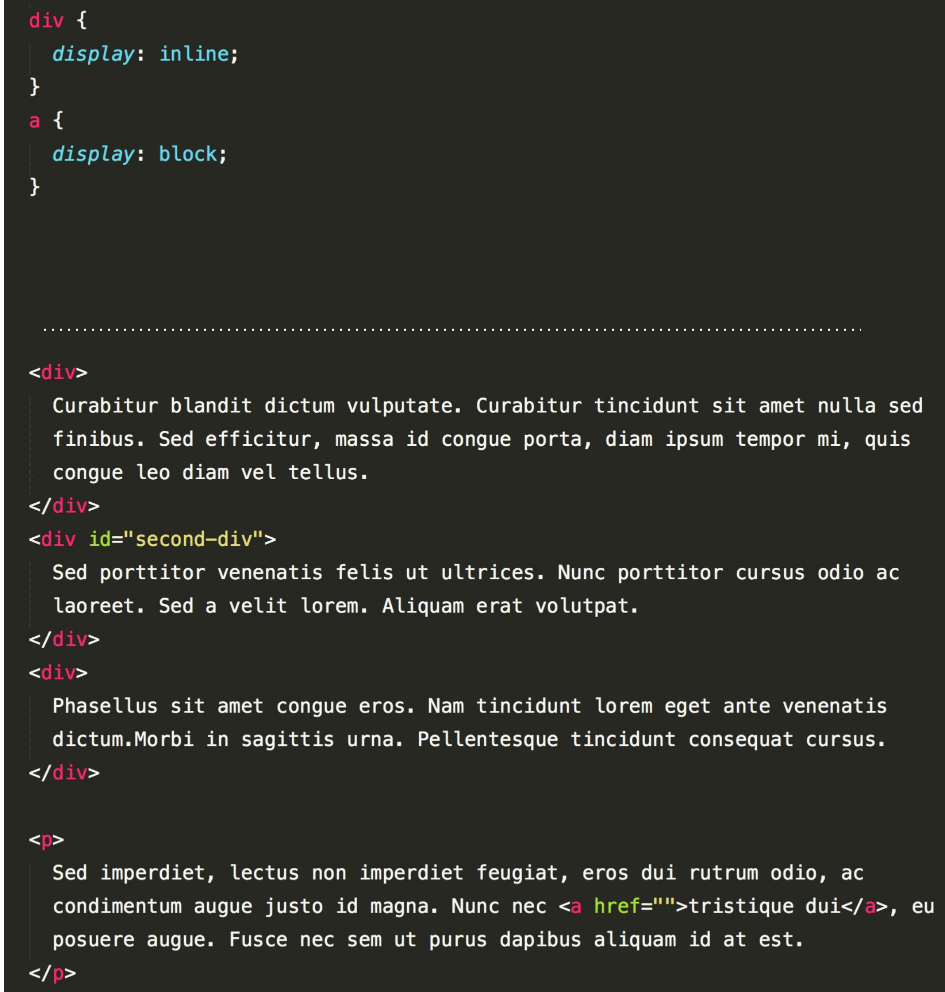


Curabitur blandit dictum vulputate. Curabitur tincidunt sit amet nulla sed finibus. Sed efficitur, massa id congue porta, diam ipsum tempor mi, quis congue leo diam vel tellus. Sed porttitor venenatis felis ut ultrices. Nunc porttitor cursus odio ac laoreet. Sed a velit lorem. Aliquam erat volutpat. Phasellus sit amet congue eros. Nam tincidunt lorem eget ante venenatis dictum.Morbi in sagittis urna. Pellentesque tincidunt consequat cursus.

Sed imperdiet, lectus non imperdiet feugiat, eros dui rutrum odio, ac condimentum augue justo id magna. Nunc nec

tristique dui

, eu posuere augue. Fusce nec sem ut purus dapibus aliquam id at est.

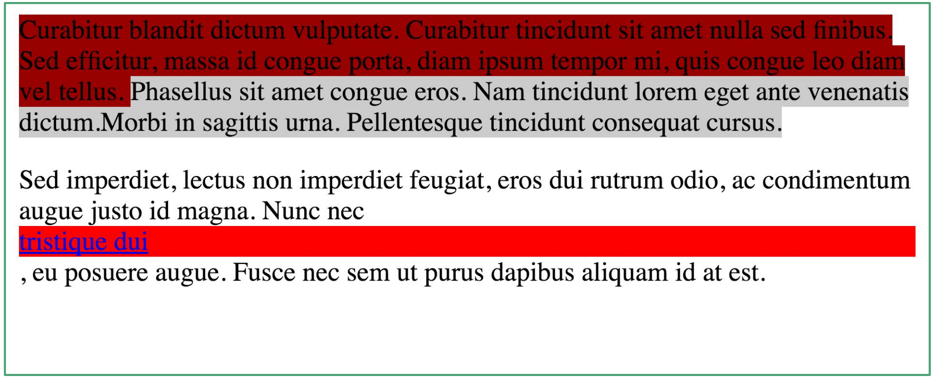


```css
div {
  display: inline;
}
a {
  display: block;
}
```

```html
<div>
  Curabitur blandit dictum vulputate. Curabitur tincidunt sit amet nulla sed
  finibus. Sed efficitur, massa id congue porta, diam ipsum tempor mi, quis
  congue leo diam vel tellus.
</div>
<div id="second-div">
  Sed porttitor venenatis felis ut ultrices. Nunc porttitor cursus odio ac
  laoreet. Sed a velit lorem. Aliquam erat volutpat.
</div>
<div>
  Phasellus sit amet congue eros. Nam tincidunt lorem eget ante venenatis
  dictum.Morbi in sagittis urna. Pellentesque tincidunt consequat cursus.
</div>

<p>
  Sed imperdiet, lectus non imperdiet feugiat, eros dui rutrum odio, ac
  condimentum augue justo id magna. Nunc nec <a href="">tristique dui</a>, eu
  posuere augue. Fusce nec sem ut purus dapibus aliquam id at est.
</p>
```

CSS display

Result:



Curabitur blandit dictum vulputate. Curabitur tincidunt sit amet nulla sed finibus. Sed efficitur, massa id congue porta, diam ipsum tempor mi, quis congue leo diam vel tellus. Phasellus sit amet congue eros. Nam tincidunt lorem eget ante venenatis dictum.Morbi in sagittis urna. Pellentesque tincidunt consequat cursus.

Sed imperdiet, lectus non imperdiet feugiat, eros dui rutrum odio, ac condimentum augue justo id magna. Nunc nec

tristique dui

, eu posuere augue. Fusce nec sem ut purus dapibus aliquam id at est.

```
div {
  display: inline;
}
a {
  display: block;
}
#second-div {
  display: none;
}
```

```
<div>
  Curabitur blandit dictum vulputate. Curabitur tincidunt sit amet nulla sed
  finibus. Sed efficitur, massa id congue porta, diam ipsum tempor mi, quis
  congue leo diam vel tellus.
</div>
<div id="second-div">
  Sed porttitor venenatis felis ut ultrices. Nunc porttitor cursus odio ac
  laoreet. Sed a velit lorem. Aliquam erat volutpat.
</div>
<div>
  Phasellus sit amet congue eros. Nam tincidunt lorem eget ante venenatis
  dictum.Morbi in sagittis urna. Pellentesque tincidunt consequat cursus.
</div>

<p>
  Sed imperdiet, lectus non imperdiet feugiat, eros dui rutrum odio, ac
  condimentum augue justo id magna. Nunc nec <a href="">tristique dui</a>, eu
  posuere augue. Fusce nec sem ut purus dapibus aliquam id at est.
</p>
```

Hiding/showing elements

CSS `visibility`

Specifies whether an element is visible.

`visible`	Default value. Visible on the page.
`hidden`	Not visible on the page. Element still takes up space on the page.

CSS `opacity`

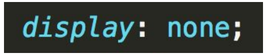
Specifies the opaqueness of an element

`1`	Default value. Visible on the page.
`0-0.99`	0 is no opacity (fully transparent). Element still takes up space on the page.

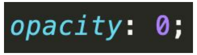
`display` vs `visibility` vs `opacity`

`display: none` – element does **not** occupy any space.

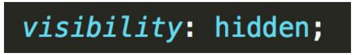
`visibility:hidden` and `opacity: 0` – element still occupies space.



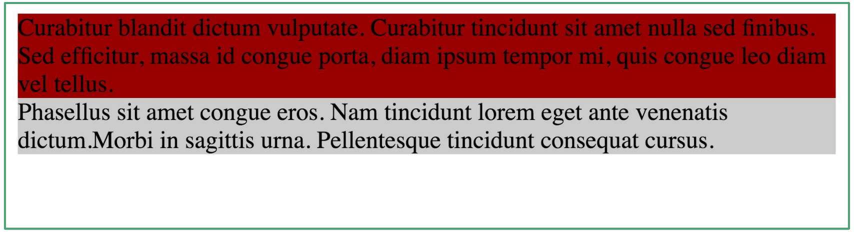
```
display: none;
```



```
opacity: 0;
```

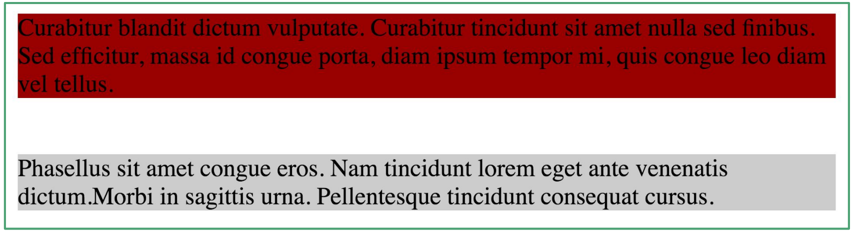


```
visibility: hidden;
```



Curabitur blandit dictum vulputate. Curabitur tincidunt sit amet nulla sed finibus. Sed efficitur, massa id congue porta, diam ipsum tempor mi, quis congue leo diam vel tellus.

Phasellus sit amet congue eros. Nam tincidunt lorem eget ante venenatis dictum.Morbi in sagittis urna. Pellentesque tincidunt consequat cursus.



Curabitur blandit dictum vulputate. Curabitur tincidunt sit amet nulla sed finibus. Sed efficitur, massa id congue porta, diam ipsum tempor mi, quis congue leo diam vel tellus.

Phasellus sit amet congue eros. Nam tincidunt lorem eget ante venenatis dictum.Morbi in sagittis urna. Pellentesque tincidunt consequat cursus.

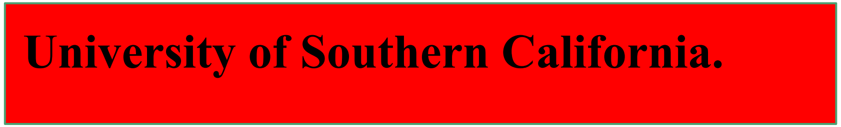
CSS Specificity

CSS specificity rules

Rule of thumb:
The more **specific** the CSS rule is, the most likely it will “win.”

If both rules are equally specific, then the rule that comes *later* “wins.”

Result:



University of Southern California.

```
#header {
    background-color: red;
}
```

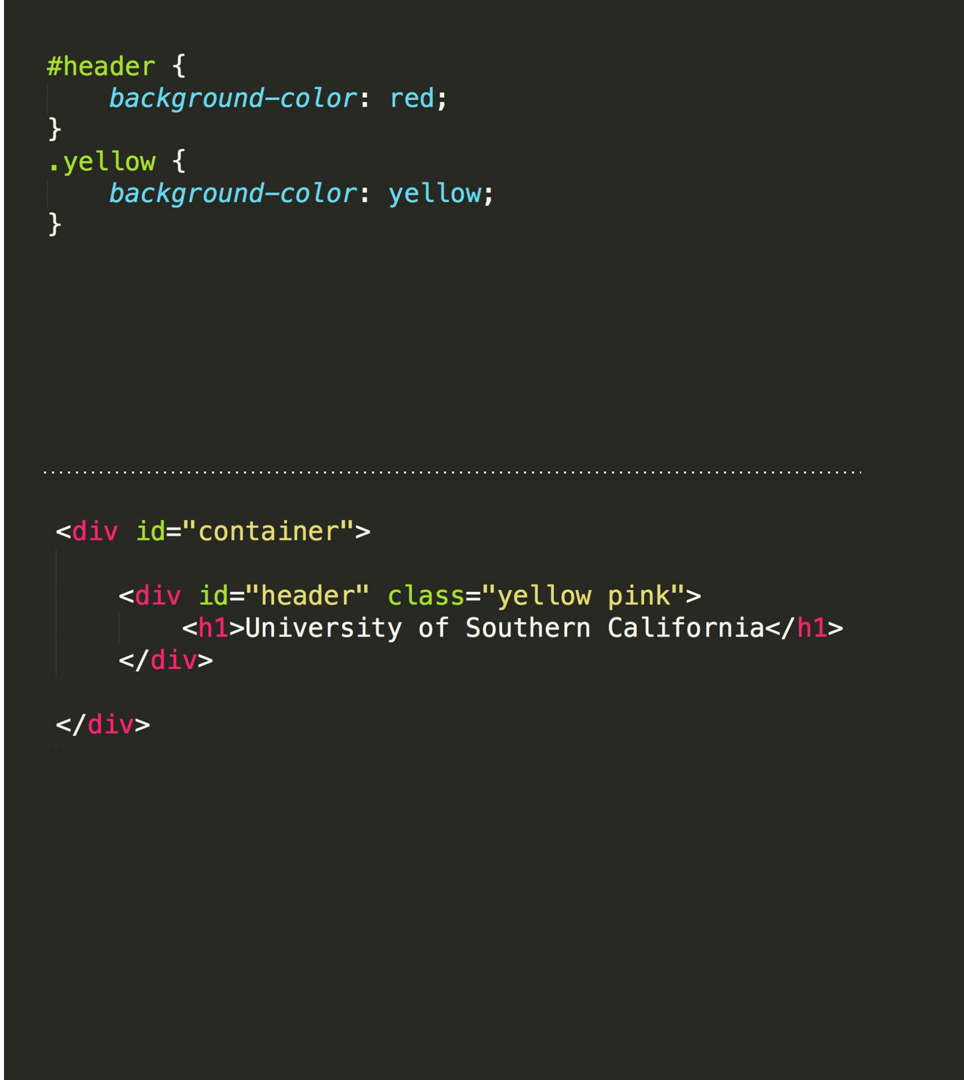
```
<div id="container">

    <div id="header" class="yellow pink">
        <h1>University of Southern California</h1>
    </div>

</div>
```

CSS specificity rules

Rule of thumb:
The more **specific** the CSS rule is, the most likely it will “win.”

If both rules are equally specific, then the rule that comes *later* “wins.”

Result:



```css
#header {
    background-color: red;
}
.yellow {
    background-color: yellow;
}
```

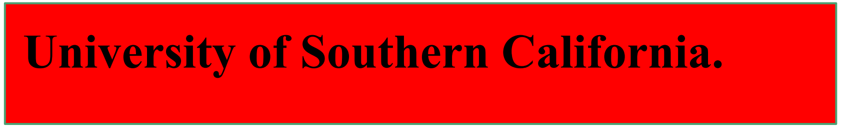
```html
<div id="container">

    <div id="header" class="yellow pink">
        <h1>University of Southern California</h1>
    </div>

</div>
```

CSS specificity rules

Rule of thumb:
The more **specific** the CSS rule is, the most likely it will “win.”

If both rules are equally specific, then the rule that comes *later* “wins.”

Result:



University of Southern California.

```css
#header {
    background-color: red;
}
.yellow {
    background-color: yellow;
}
```

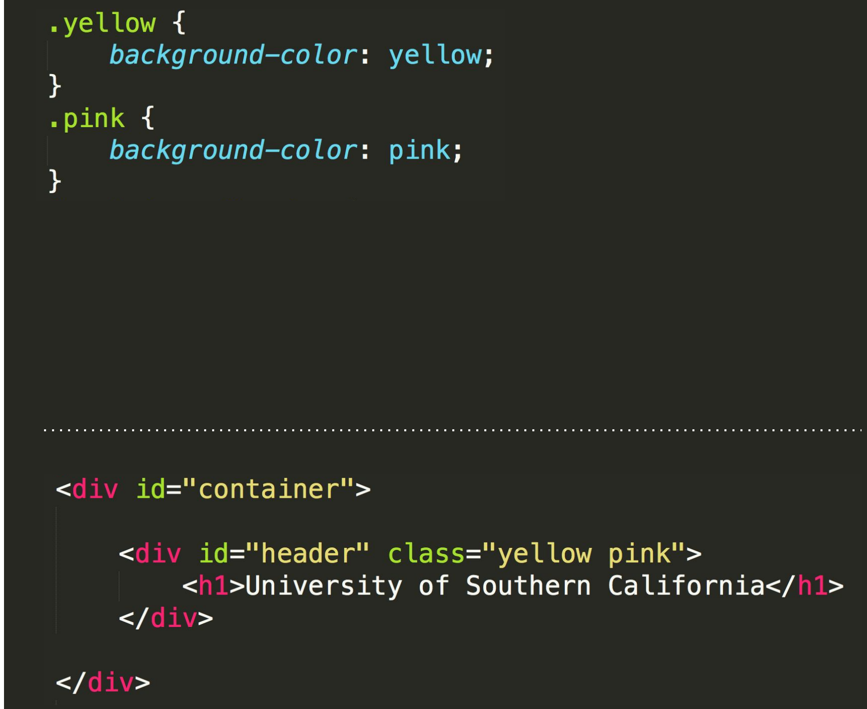
```html
<div id="container">

    <div id="header" class="yellow pink">
        <h1>University of Southern California</h1>
    </div>

</div>
```

CSS specificity rules

Rule of thumb:
The more **specific** the CSS rule is, the most likely it will “win.”

If both rules are equally specific, then the rule that comes *later* “wins.”

Result:



```css
.yellow {
    background-color: yellow;
}
.pink {
    background-color: pink;
}
```

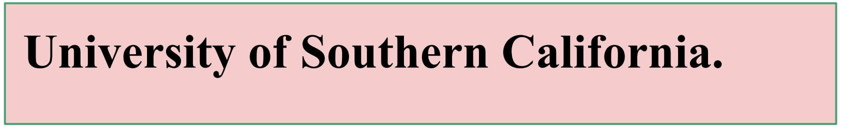
```html
<div id="container">

    <div id="header" class="yellow pink">
        <h1>University of Southern California</h1>
    </div>

</div>
```

CSS specificity rules

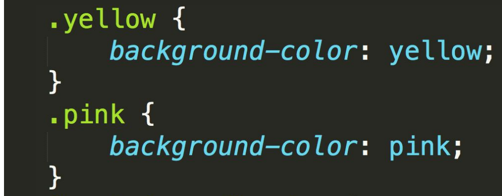
Rule of thumb:
The more **specific** the CSS rule is, the most likely it will “win.”

If both rules are equally specific, then the rule that comes *later* “wins.”

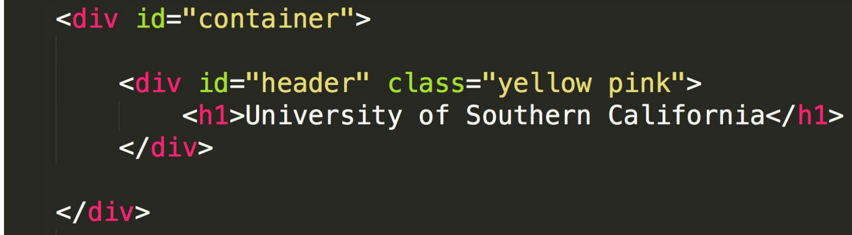
Result:



University of Southern California.



```
.yellow {
    background-color: yellow;
}
.pink {
    background-color: pink;
}
```



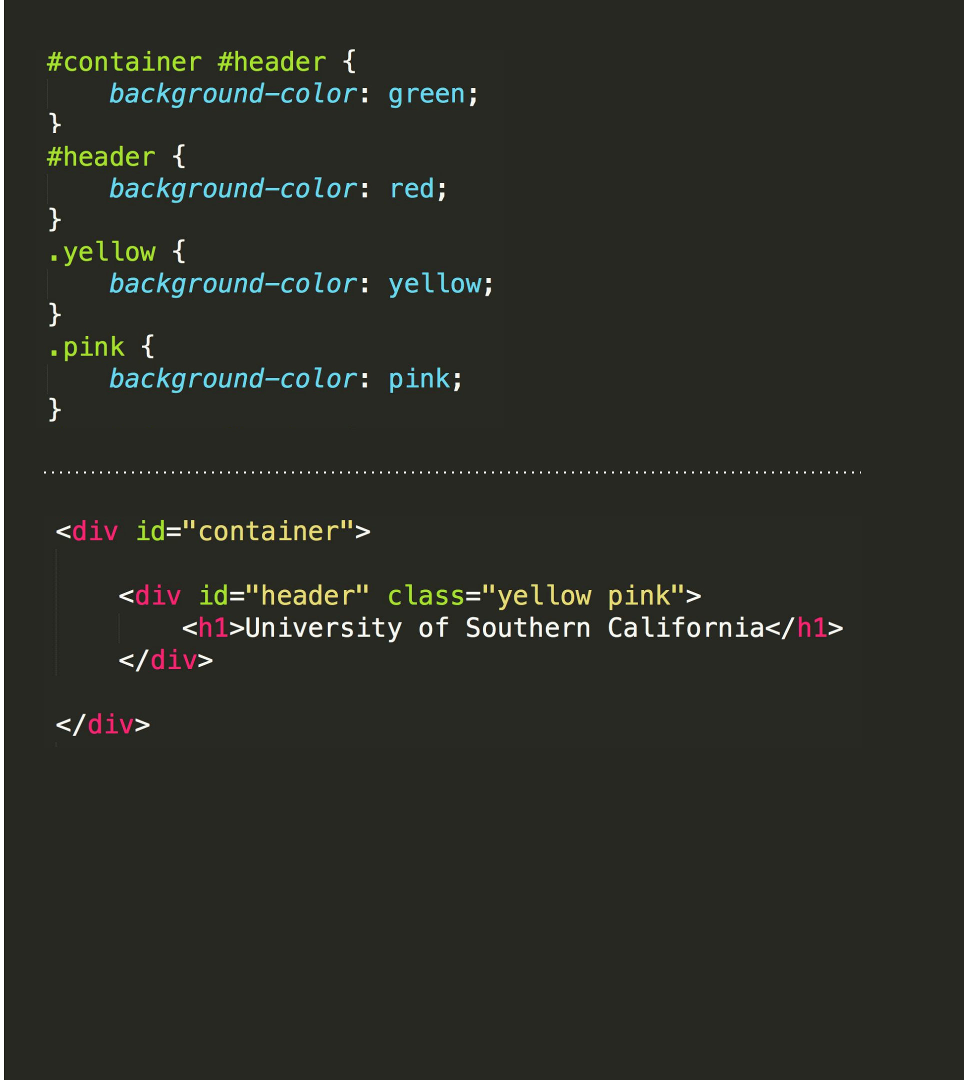
```
<div id="container">

    <div id="header" class="yellow pink">
        <h1>University of Southern California</h1>
    </div>

</div>
```

CSS specificity rules

Rule of thumb:
The more **specific** the CSS rule is, the most likely it will “win.”

If both rules are equally specific, then the rule that comes *later* “wins.”

Result:



```
#container #header {
    background-color: green;
}
#header {
    background-color: red;
}
.yellow {
    background-color: yellow;
}
.pink {
    background-color: pink;
}
```

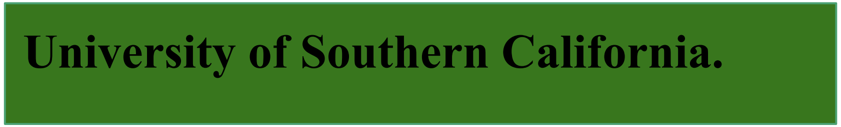
```
<div id="container">

    <div id="header" class="yellow pink">
        <h1>University of Southern California</h1>
    </div>

</div>
```

CSS specificity rules

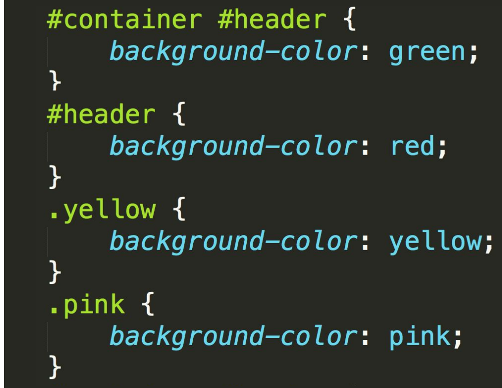
Rule of thumb:
The more **specific** the CSS rule is, the most likely it will “win.”

If both rules are equally specific, then the rule that comes *later* “wins.”

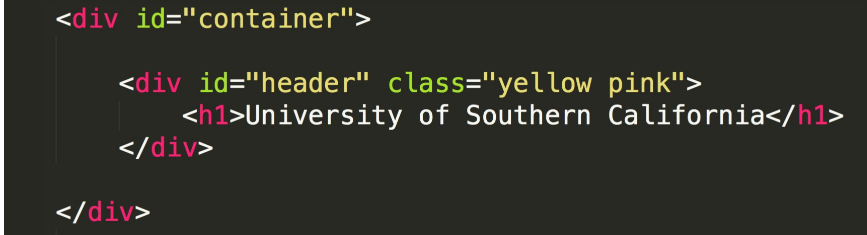
Result:



University of Southern California.



```css
#container #header {
    background-color: green;
}
#header {
    background-color: red;
}
.yellow {
    background-color: yellow;
}
.pink {
    background-color: pink;
}
```



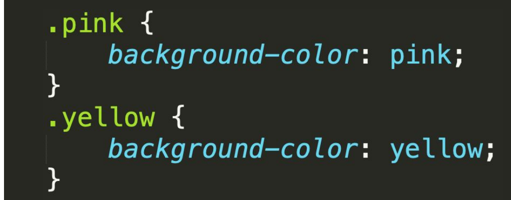
```html
<div id="container">

    <div id="header" class="yellow pink">
        <h1>University of Southern California</h1>
    </div>

</div>
```

CSS specificity rules

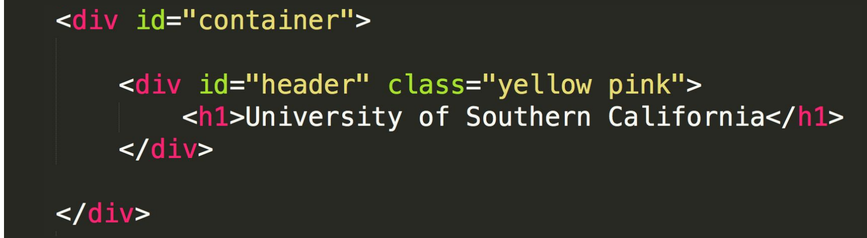
Rule of thumb:
The more **specific** the CSS rule is, the most likely it will “win.”

If both rules are equally specific, then the rule that comes *later* in the **stylesheet** “wins.”

Result:



```
.pink {
    background-color: pink;
}
.yellow {
    background-color: yellow;
}
```



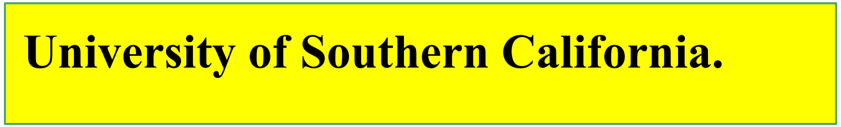
```
<div id="container">

    <div id="header" class="yellow pink">
        <h1>University of Southern California</h1>
    </div>

</div>
```

CSS specificity rules

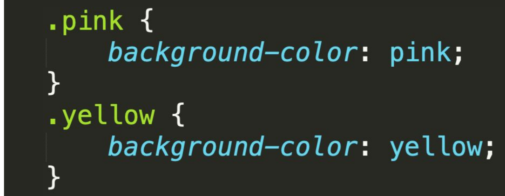
Rule of thumb:
The more **specific** the CSS rule is, the most likely it will “win.”

If both rules are equally specific, then the rule that comes *later* in the **stylesheet** “wins.”

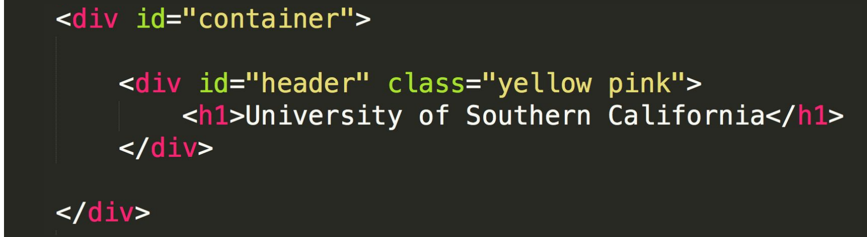
Result:



University of Southern California.



```css
.pink {
    background-color: pink;
}
.yellow {
    background-color: yellow;
}
```



```html
<div id="container">

    <div id="header" class="yellow pink">
        <h1>University of Southern California</h1>
    </div>

</div>
```

Responsive Web Design

CONTENT IS LIKE WATER

> " You put water into a cup it becomes the cup.
> You put water into a bottle it becomes the bottle.
> You put it in a teapot, it becomes the teapot. "

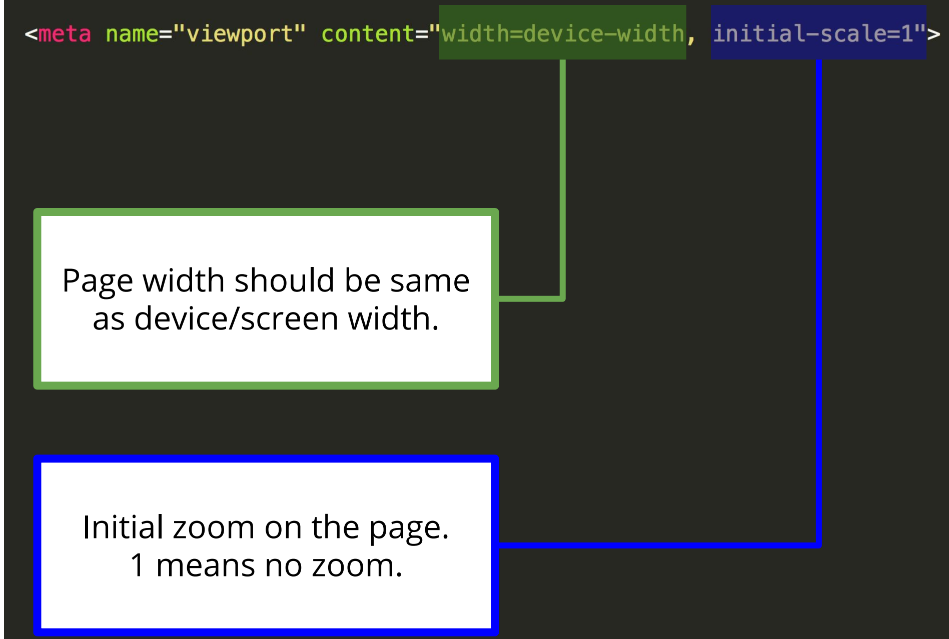
Josh Clark *(originally Bruce Lee)* - Seven deadly mobile myths

Illustration by Stéphanie Walter

Viewport Meta Tag

Specifies how viewport should behave.

Must be included in `<head>` tag of every responsive site.



```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Page width should be same as device/screen width.

Initial zoom on the page. 1 means no zoom.

CSS `@media` rule

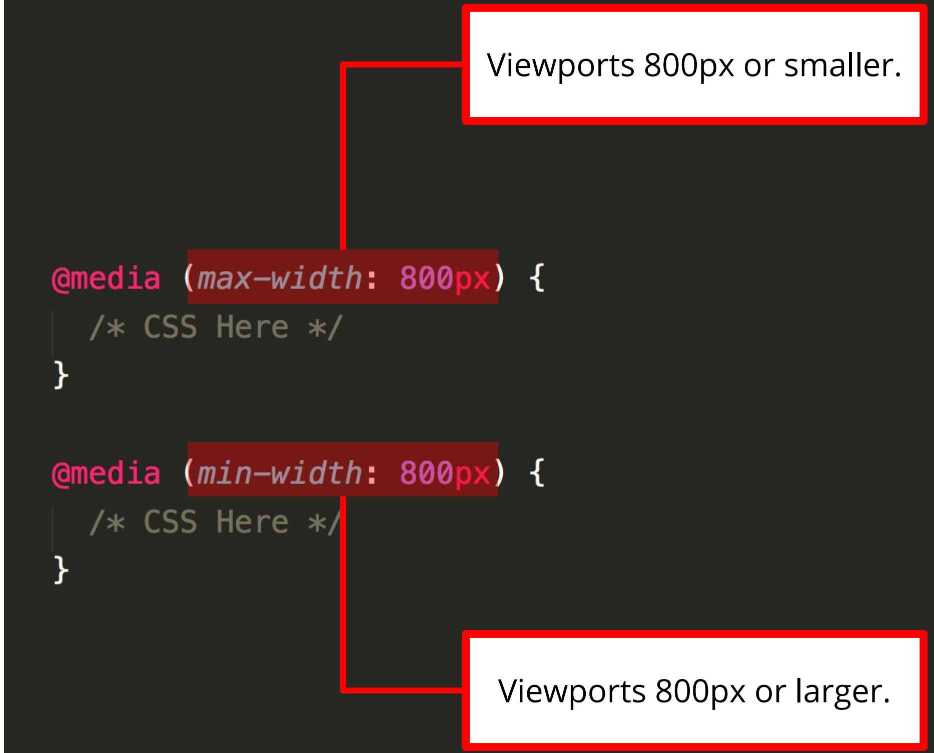
Also called **media query**.

Specifies different CSS rules for different devices.

No set rules or standard breakpoints.

Breakpoints used by Bootstrap:

≥ 1200px	Large Desktops
1199px - 992px	Desktops / Laptops
991px - 768px	Tablets
767px - 576px	Landscape Phones
≤ 575px	Portrait Phones



```
@media (max-width: 800px) {
  /* CSS Here */
}

@media (min-width: 800px) {
  /* CSS Here */
}
```

Viewports 800px or smaller.

Viewports 800px or larger.

CSS `min-width, max-width`

Sets the minimum or maximum width of an element. Useful in creating responsive elements; may reduce need to create media queries.

`min-width`	Prevents the used value of the width property from becoming *smaller* than the value specified for min-width.
`max-width`	Prevents the used value of the width property from becoming l*arger* than the value specified for max-width.

Desktop First Responsive Design

Extra Large Devices

(Large Desktops)

Large Devices

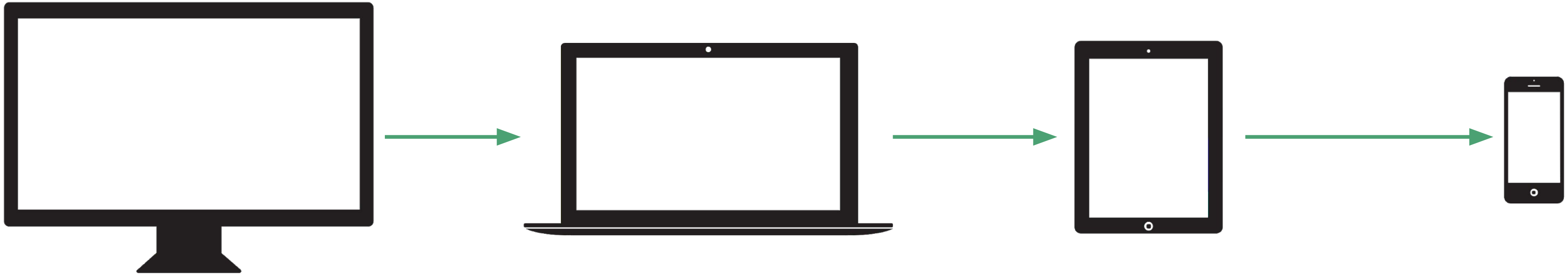
(Desktops/Laptops)

Medium Devices

(Tablets)

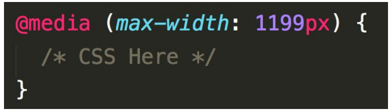
Small Devices

(Mobile Phones)

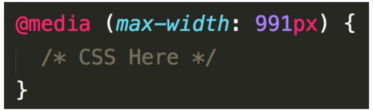


Default CSS.

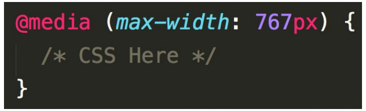
No media query.



```
@media (max-width: 1199px) {
  /* CSS Here */
}
```



```
@media (max-width: 991px) {
  /* CSS Here */
}
```



```
@media (max-width: 767px) {
  /* CSS Here */
}
```

Mobile First Responsive Design

Small Devices

(Mobile Phones)

Medium Devices

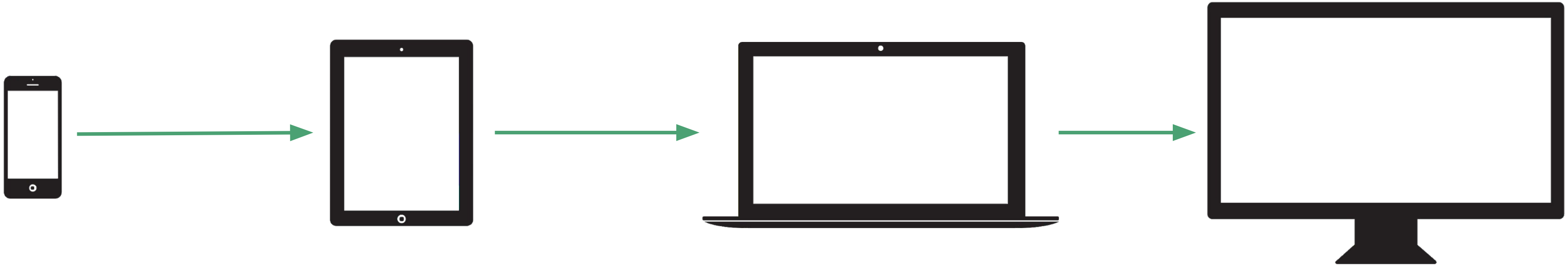
(Tablets)

Large Devices

(Desktops/Laptops)

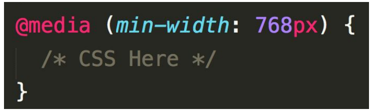
Extra Large Devices

(Large Desktops)

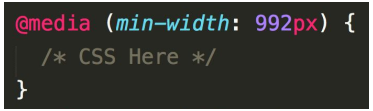


Default CSS.

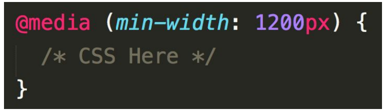
No media query.



```css
@media (min-width: 768px) {
  /* CSS Here */
}
```



```css
@media (min-width: 992px) {
  /* CSS Here */
}
```



```css
@media (min-width: 1200px) {
  /* CSS Here */
}
```

Fonts

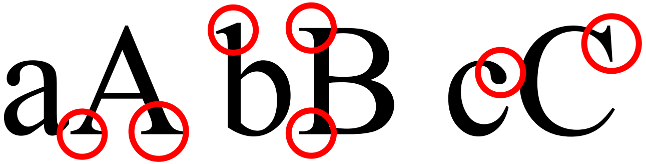
Typefaces (Font Families)

4 popular web typefaces:

Serif	Fonts with small lines (serifs) attached at the end of strokes in letters.
Sans-Serif	Fonts without serifs (small lines).
Monospace	Fonts with letters & characters each occupying same amount of horizontal space.
Cursive	*Fonts that emulate handwriting.*

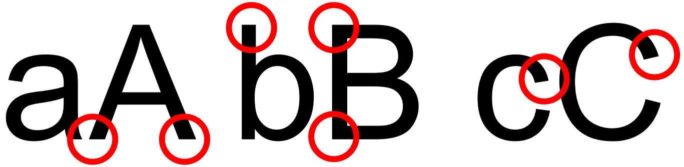
Serif vs Sans-Serif Typefaces

Serif
(Times New Roman)



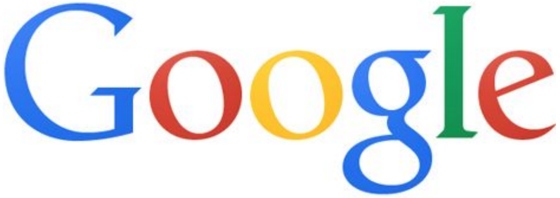
aA bB cC

Sans-Serif
(Arial)

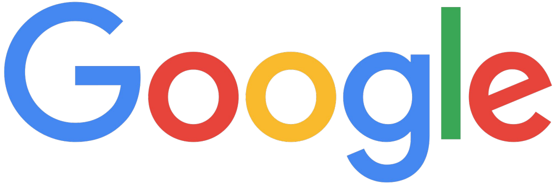


aA bB cC

Serif vs Sans-Serif Typefaces

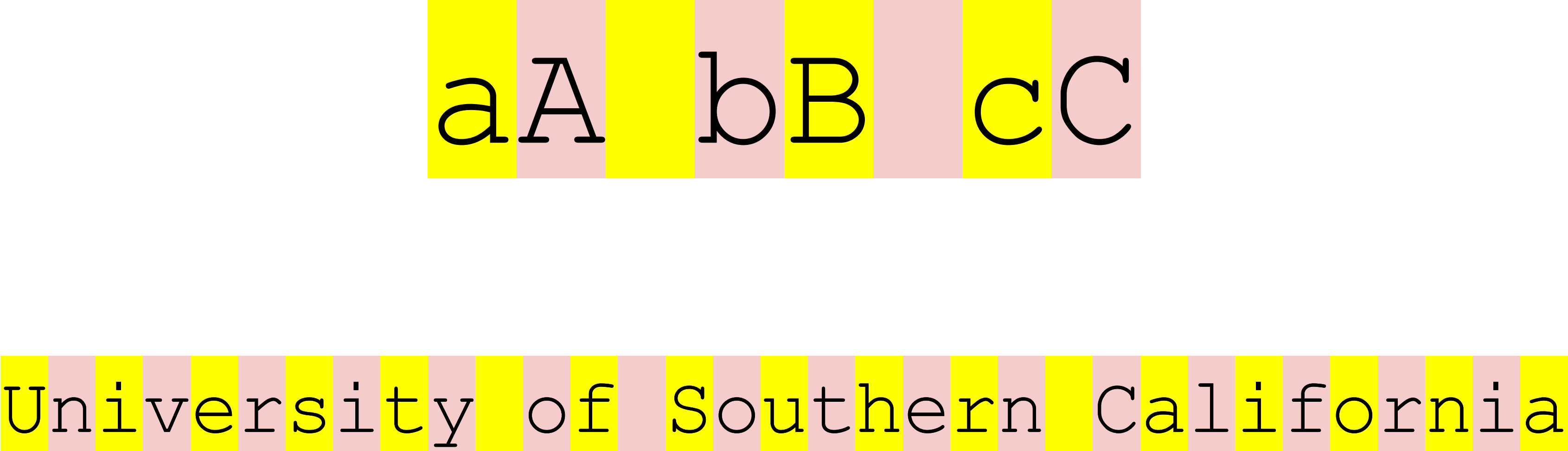


2013 - 2015



2015 - Present

Monospace Typeface



aA bB cC

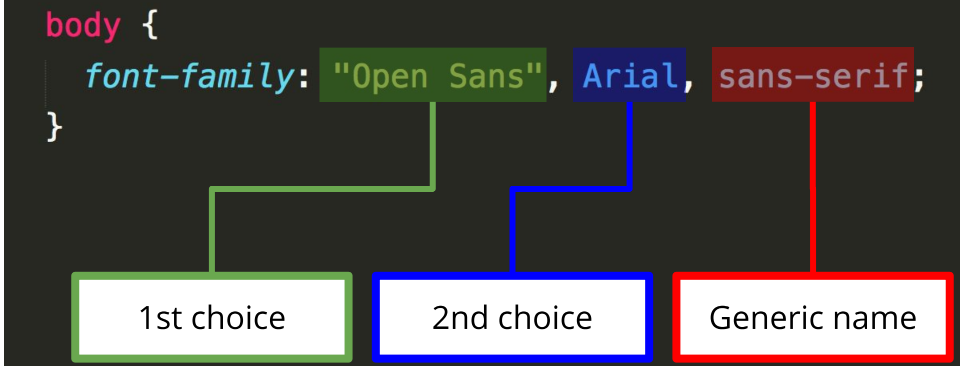
University of Southern California

CSS `font-family`

Specifies typefaces to be applied in prioritized order.

Always include generic typeface at the end.

Use quotations for font names with more than 1 word.



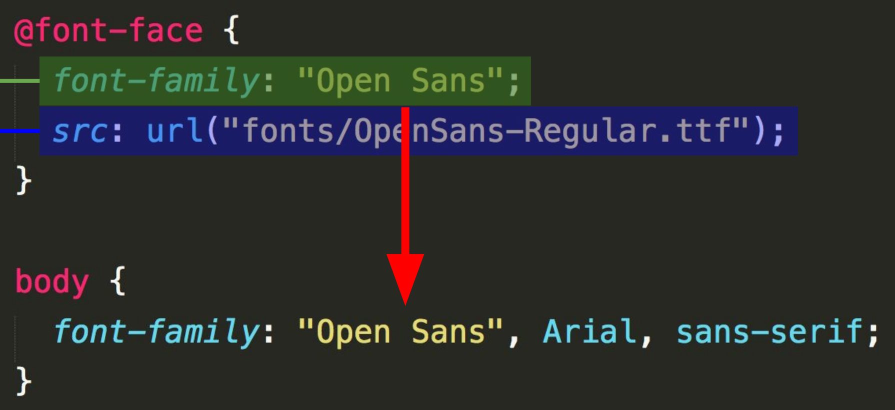
```css
body {
  font-family: "Open Sans", Arial, sans-serif;
}
```

CSS `@font-face` rule

Loads custom fonts.

Required descriptors:

1. Font name,
2. Font location.



```
@font-face {
  font-family: "Open Sans";
  src: url("fonts/OpenSans-Regular.ttf");
}

body {
  font-family: "Open Sans", Arial, sans-serif;
}
```

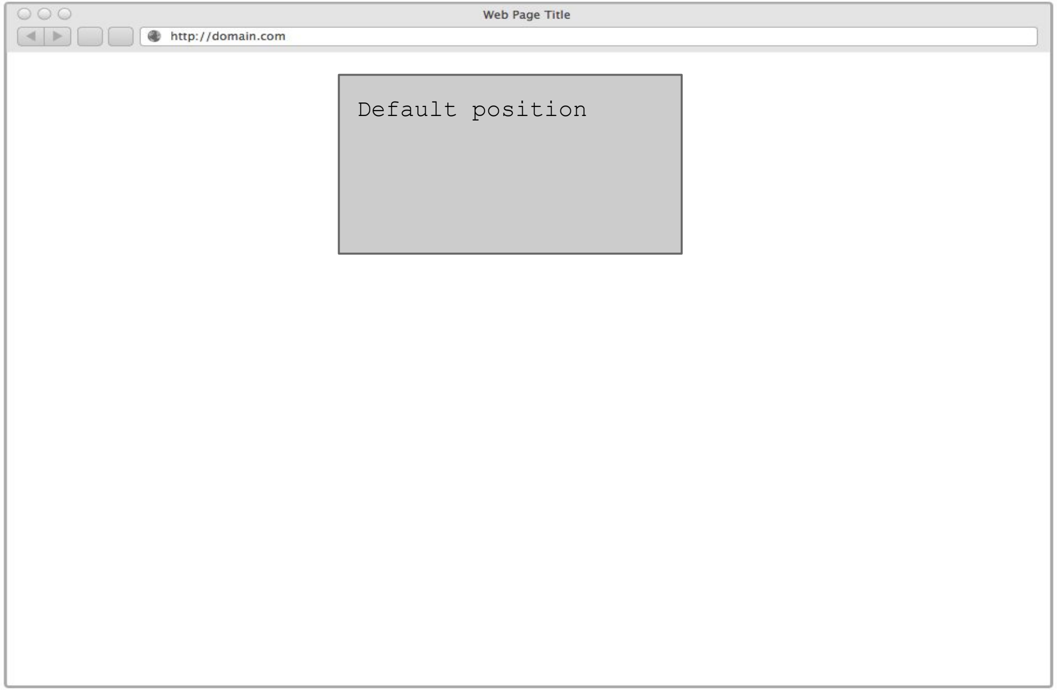
Positions

CSS `position`

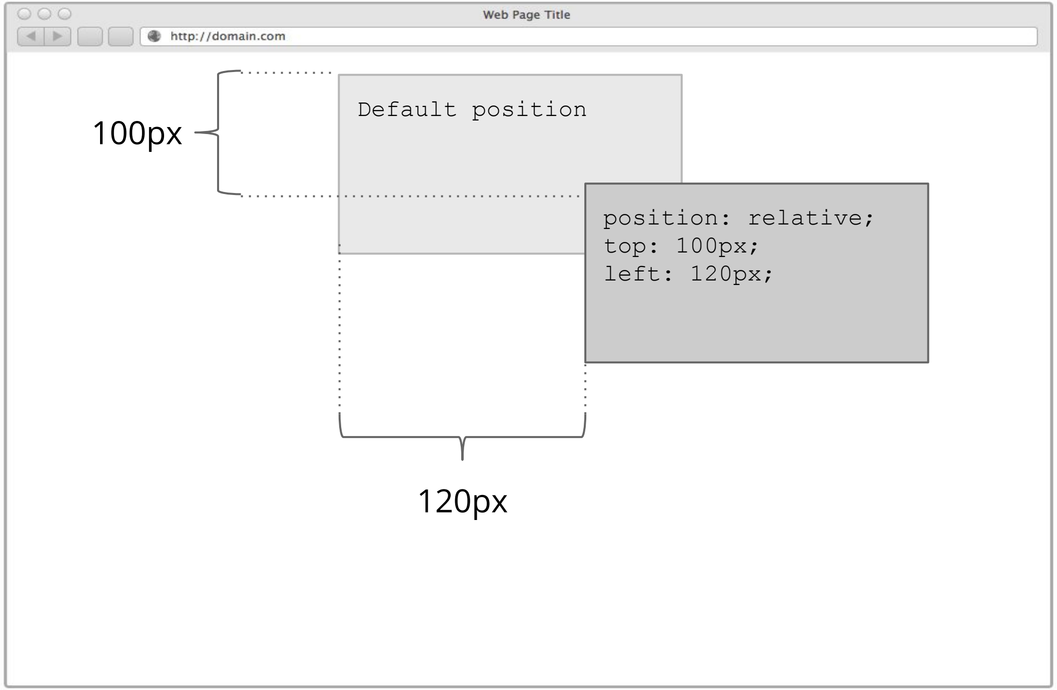
Lifts elements out of the “normal flow” of the document and positions it accordingly.

`static`	Default value. Positioned according to normal flow. `top`, `right`, `bottom`, `left`, `z-index` do not work.
`fixed`	Positioned relative to the browser, even when scrolled. Does not occupy space within normal flow.
`relative`	Positioned with respect to original position.
`absolute`	Positioned with respect to closest ancestor with position **not** `static`. Does not occupy space within normal flow.

CSS `position: relative`

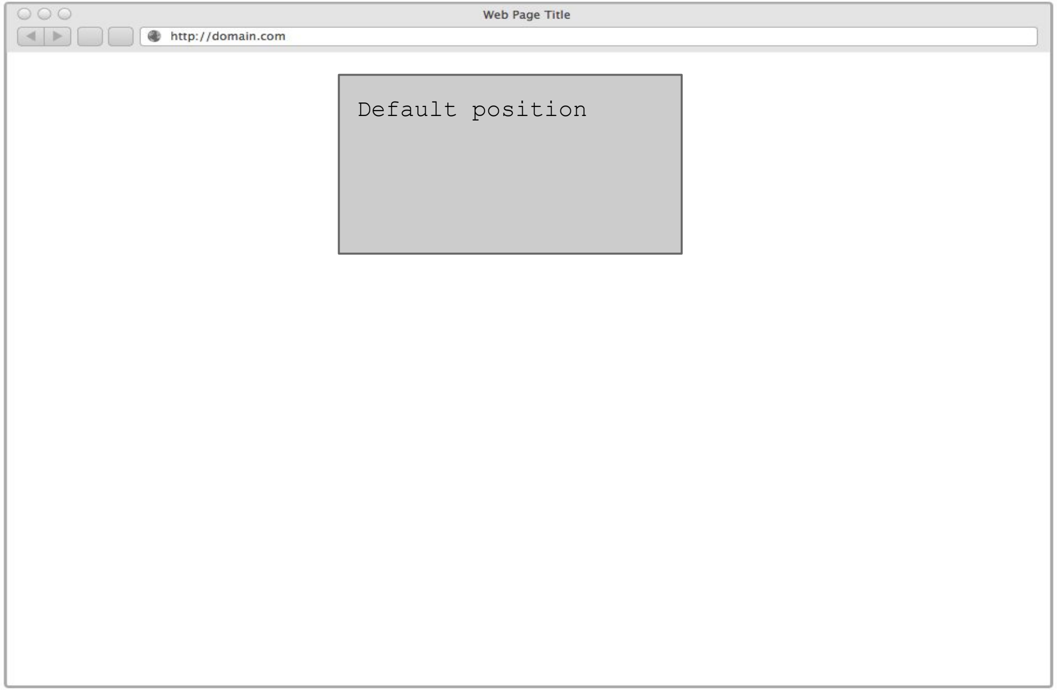


CSS `position: relative`

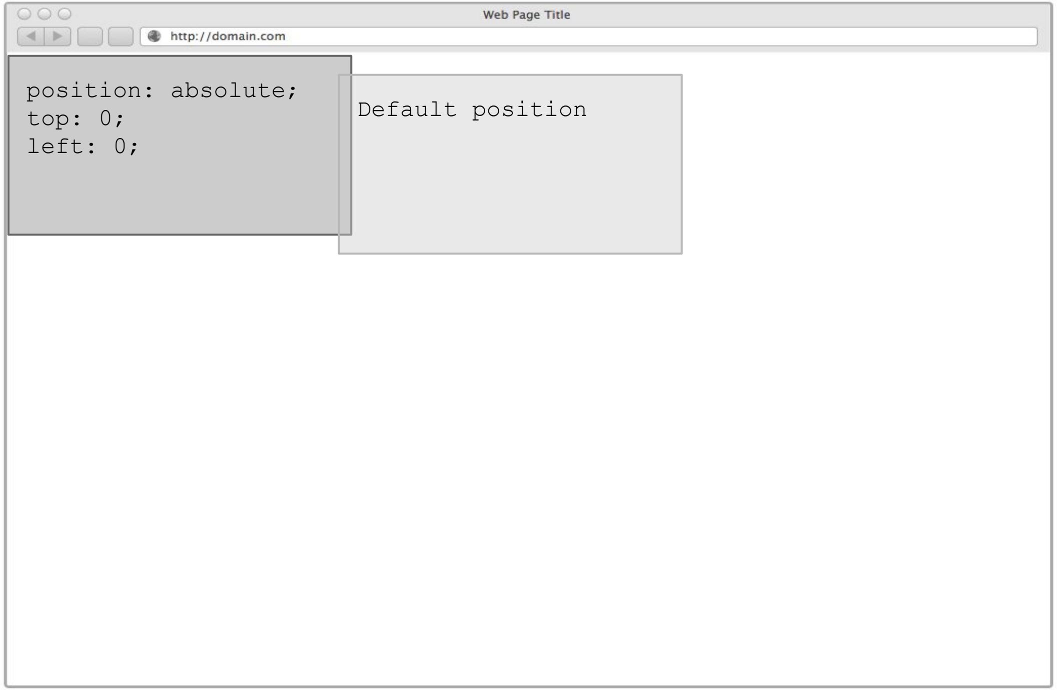


```
position: relative;
top: 100px;
left: 120px;
```

CSS `position: absolute`



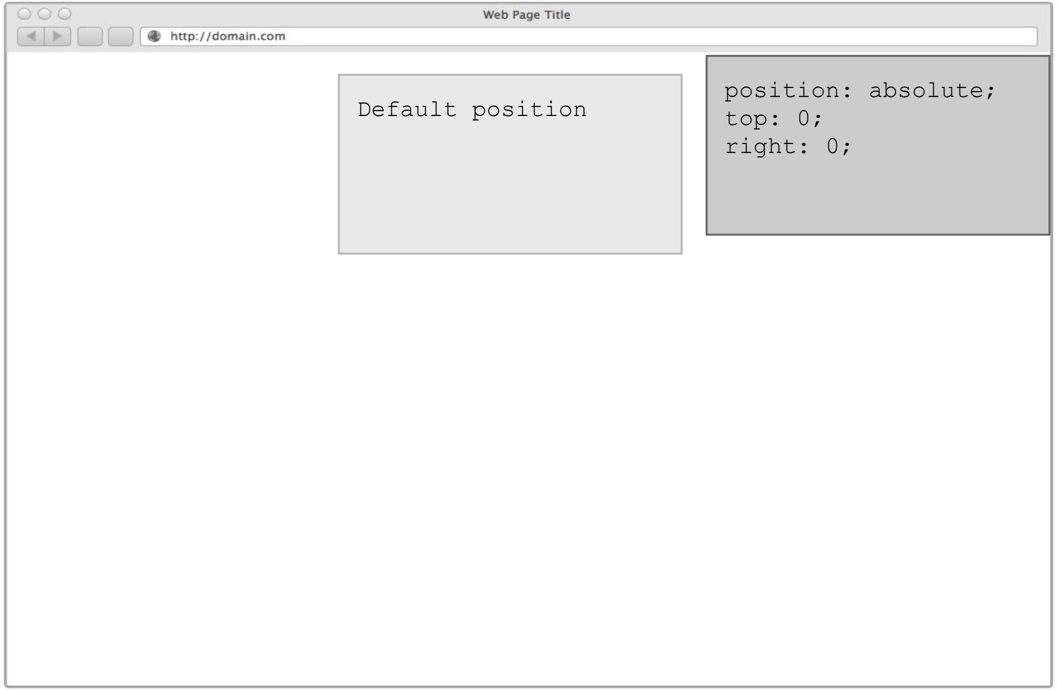
CSS `position: absolute`



```
position: absolute;
top: 0;
left: 0;
```

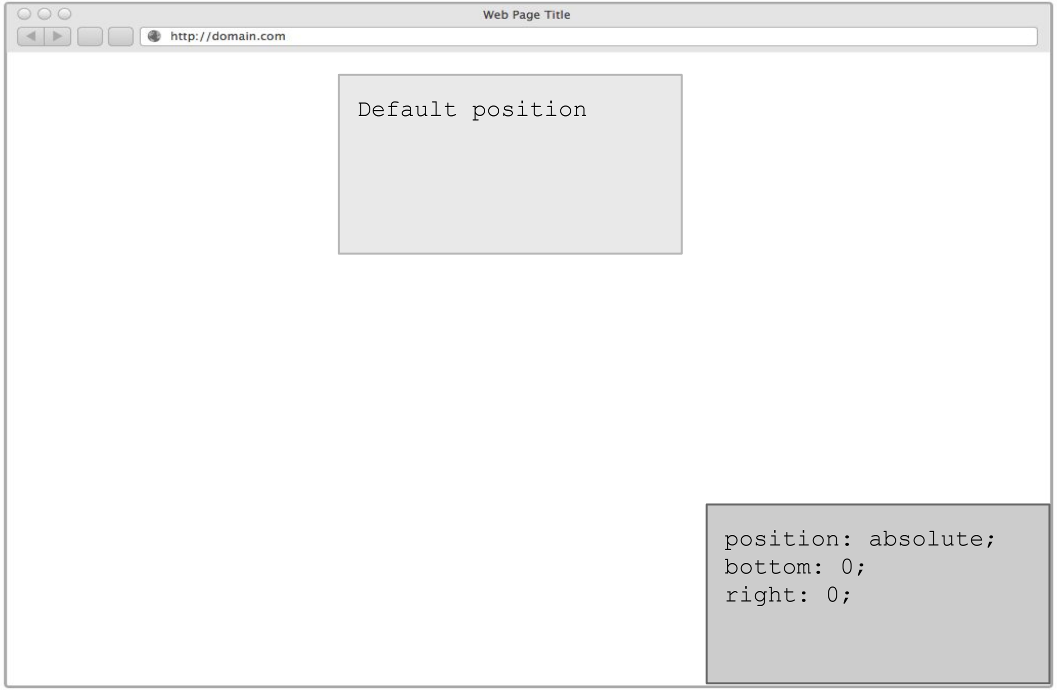
Default position

CSS `position: absolute`



```
position: absolute;
top: 0;
right: 0;
```

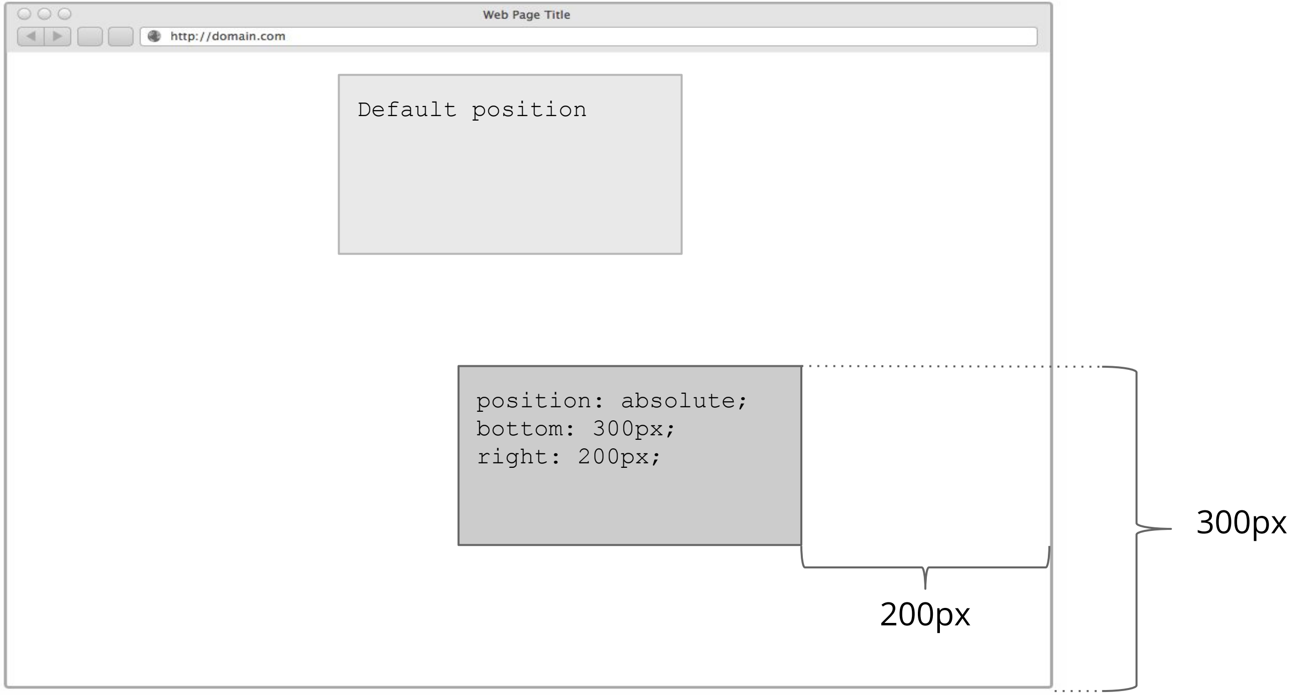
CSS `position: absolute`



Default position

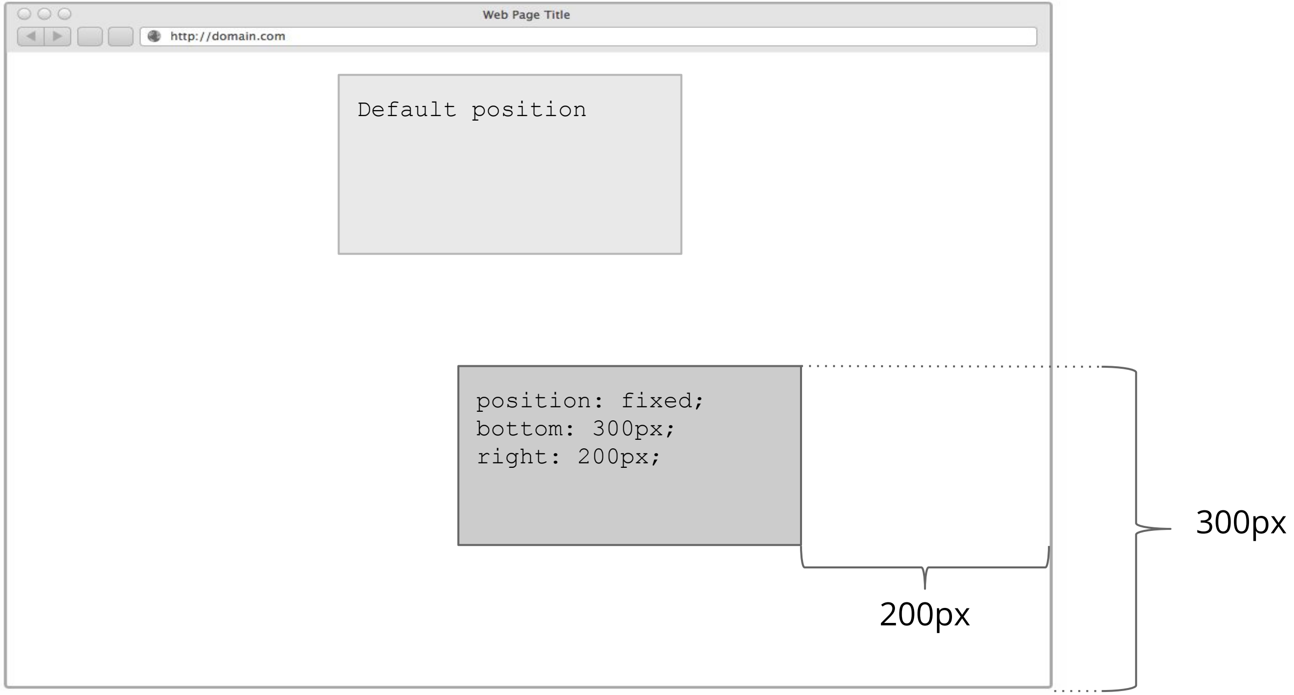
```
position: absolute;
bottom: 0;
right: 0;
```

CSS `position: absolute`



```
position: absolute;
bottom: 300px;
right: 200px;
```

CSS `position: fixed`


Web Page Title
http://domain.com
Default position
position: fixed;
bottom: 300px;
right: 200px;
300px
200px