Internet Technologies

Introduction to HTML and CSS - Part 3
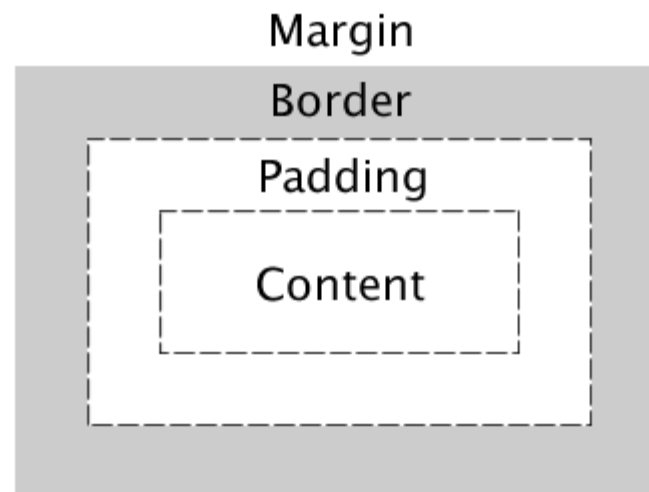
Box Model, Positioning

University of Cyprus
Department of Computer Science

The CSS Box Model

- Every HTML element is composed of 4 layers:
  - the element's content
  - the **border** around the element's content
  - **padding** space between the content and border (inside)
  - a **margin** clears the area around border (outside)

border

- Specify border (3 properties can be set – width, style, color – order does not matter)
  - border: solid;
  - border: dashed red;
  - border: 10px solid;
  - border: 5px double #32a1ce;
- Can also specify each border individually:
  - border-top
  - border-bottom
  - border-left
  - border-right

This is a box with a border around it.

This is a box with a border around it.

This is a box with a border around it.

This is a box with a border around it.

border: width style color;

```
<p>Welcome to CS425: Internet Technologies!
In this course you will learn modern web
development techniques.</p>
```

```
p {
    border: 2px solid black;
}
```

Welcome to CS425: Internet Technologies! In this course you will learn
modern web development techniques.

Rounded border

- Can specify the border-radius to make rounded corners:

    `border-radius: 10px;`

- You don't actually need to set a border to use border-radius.

```css
p {
    background-color: purple;
    border-radius: 10px;
    color: white;
}
```

Welcome to CS425: Internet Technologies! In this course you will learn modern web development techniques.

Borders look a little squished

- When we add a border to an element, it sits flush against the text:

Welcome to CS425:
modern web develo

- Q: How do we add space between the border and the content of the element?

padding

```css
p {
  border: 2px solid black;
  padding: 10px;
}
```
CSS

Welcome to CS425: Internet Technologies! In this course you will learn modern web development techniques.

- padding is the space between the border and the content.
  - o Can specify padding-top, padding-bottom, padding-left, padding-right
  - o There's also a shorthand (clockwise starting from top):
    ```
    padding: 2px 4px 3px 1px;    <- top|right|bottom|left
    padding: 10px 2px;           <- top&bottom|left&right
    padding: 1px 2px 3px;        <- top|left&right|bottom
    ```
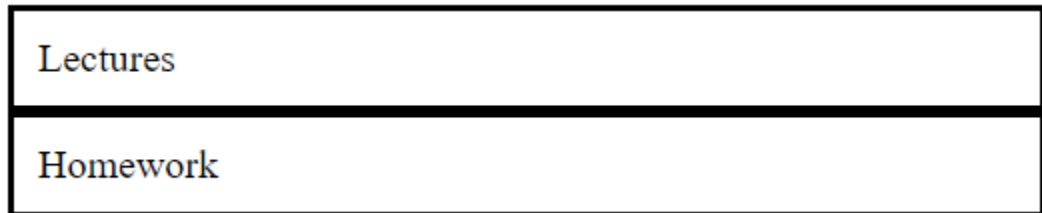
`<div>`s look a little squished

- When we add a border to multiple `div`s, they sit flush against each other:

```html
<div> Lectures </div>
<div> Homework </div>
```
HTML

```css
div {
    border: 2px solid black;
    padding: 10px;
}
```
CSS

Lectures
Homework

- Q: How do we add space between multiple elements?

`<div>`s look a little squished

- When we add a border to multiple `div`s, they sit flush against each other:

```html
<div> Lectures </div>          HTML
<div> Homework </div>
```

```css
div {                          CSS
    border: 2px solid black;
    padding: 10px;
}
```
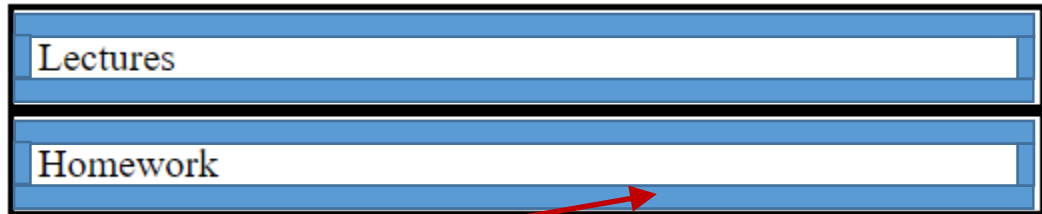

- Q: How do we add space between multiple elements?

margin

```css
div {
    margin: 20px;
    border: 2px solid black;
    padding: 10px;
}
```
CSS

Lectures

Homework

- margin is the space between the border and other elements..
  - o Can specify margin-top, margin-bottom, margin-left, margin-right
  - o There's also a shorthand (clockwise starting from top):
    margin: 2px 4px 3px 1px;   <- top|right|bottom|left
    margin: 10px 2px;   <- top&bottom|left&right
    margin: 1px 2px 3px;   <- top|left&right|bottom

margin

- Actually, why doesn't this:

```css
div {
    margin: 20px;
    border: 2px solid black;
    padding: 10px;
}
```
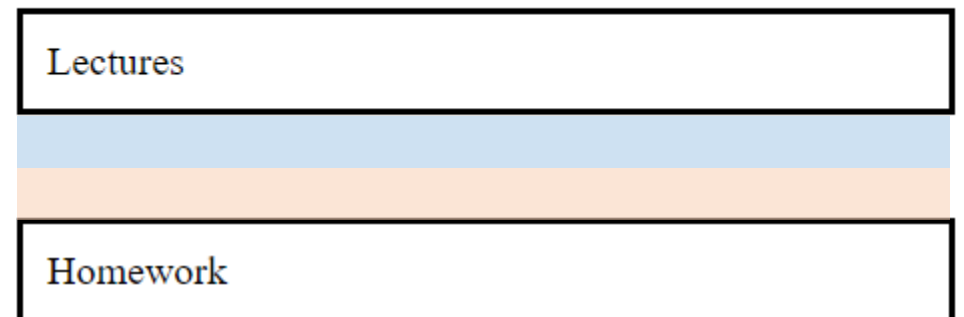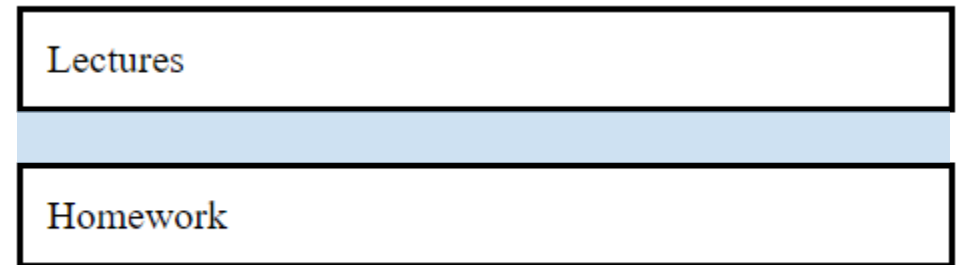
- Look more like this?

margin

- Actually, why doesn't this:

```css
div {
    margin: 20px;
    border: 2px solid black;
    padding: 10px;
}
```
CSS

- Look more like this?

20px `margin-bottom`
+ 20px `margin-top` =
40px margin ?

margin collapsing

- Sometimes the top and bottom margins of block elements are combined ("collapsed") into a single margin.
  - This is called **margin collapsing**

- Generally if:
  - The elements are siblings
  - The elements are block-level (**not** inline-block)

- then they collapse into **max**(`margin-bottom`, `margin-top`).

- (There are [some exceptions](#) to this, but when in doubt, use the Page Inspector to see what's going on.)

Negative margin

- Margins **can be** negative as well.
  - ○ Example: negative margin on image:

```html
<div id="header"></div>
<div id="profile">
    <img src="https://www.cs.ucy.ac.cy/images/ucy_logo_en.png" />
</div>
```

```css
#header {
    background-color: lightblue;
    height: 200px;
}
img {
    /* initially no negative margin*/
    margin-left: 50px;
    height: 65px;
    border: 2px solid lightgray;
}
```

Negative margin

- Margins **can be** negative as well.
  - ○ `img { margin-top: -40px; }`

```html
<div id="header"></div>
<div id="profile">
    <img src="https://www.cs.ucy.ac.cy/images/ucy_logo_en.png" />
</div>
```

```css
#header {
    background-color: lightblue;
    height: 200px;
}
img {
    margin-top: -40px;
    margin-left: 50px;
    height: 65px;
    border: 2px solid lightgray;
}
```

auto margins

- If you set `margin-left` and `margin-right` to `auto`, you can horizontally center a block-level element with a specified width:

```
                                        HTML
<div>
    This is a box of text
</div>
```

```
                                        CSS
div {
    margin-left: auto;
    margin-right: auto;
    border: 2px solid black;
    padding: 10px;
    width: 300px;
}
```
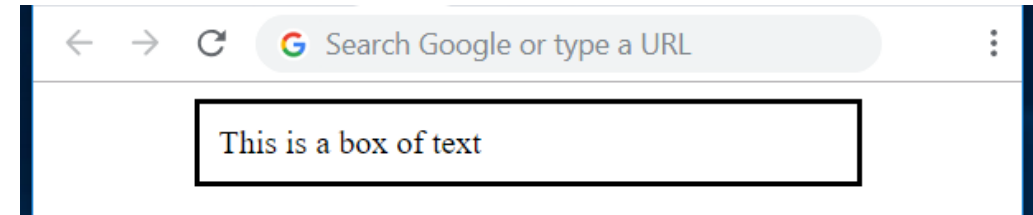
Moving things with `position`

- **Positioned layout** lets you define precisely where an element should be in the page ([mdn](#)).

- You can use positioned layout doing the following:

  1. Define a **position** method:
     `static`, `fixed`, `absolute`, `relative`

  2. Define **offsets**:
     `top`, `left`, `bottom`, `right`

  3. (optional) Define **z-index** for overlapping layers ([mdn](#))

Moving things with `position`

- To specify exactly where an element goes, set its **top**, **left**, **bottom**, and/or **right** offset.

- The meaning of these offset values depend on the reference point set by position:
  - **static**: no reference point; static block can't move (this is the default style for every element)
  - **fixed**: a fixed position within the viewport
  - **absolute**: a fixed position within its "containing element"
  - **relative**: offset from its normal static position

position: static
(nothing happens!)

- static is the default value for position

- If you use top/left/bottom/right without
setting non-static position, nothing will happen

```html
<h1>Puppy</h1>
<p>A puppy is a juvenile dog. Some puppies can
weigh 1-3 lb (0.45-1.36 kg), while larger ones
can weigh up to 15-23 lb (6.8-10.4 kg).</p>
<h2>Development</h2>
<p>At first, puppies spend the large
of their time sleeping and the rest
feeding.</p>
<div id="box1"></div>
```
HTML

```css
body {
    font-family: Helvetica, sans-serif;
}
#box1 {
    height: 100px;
    width: 100px;
    background-color: red;
}
```

Puppy

A puppy is a juvenile dog. Some puppies can weigh 1–3 lb (0.45–1.36 kg), while larger ones can weigh up to 15–23 lb (6.8–10.4 kg).

Development

At first, puppies spend the large majority of their time sleeping and the rest feeding.
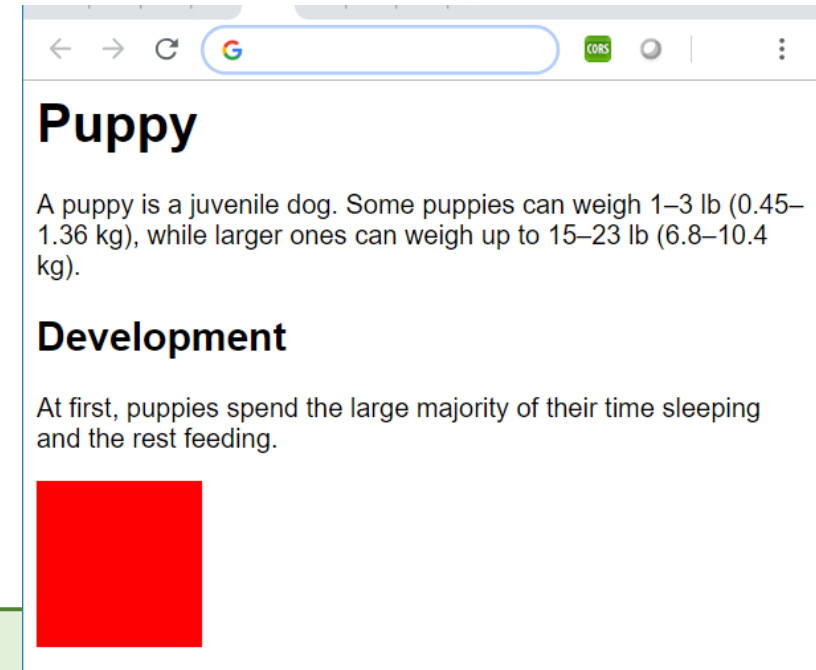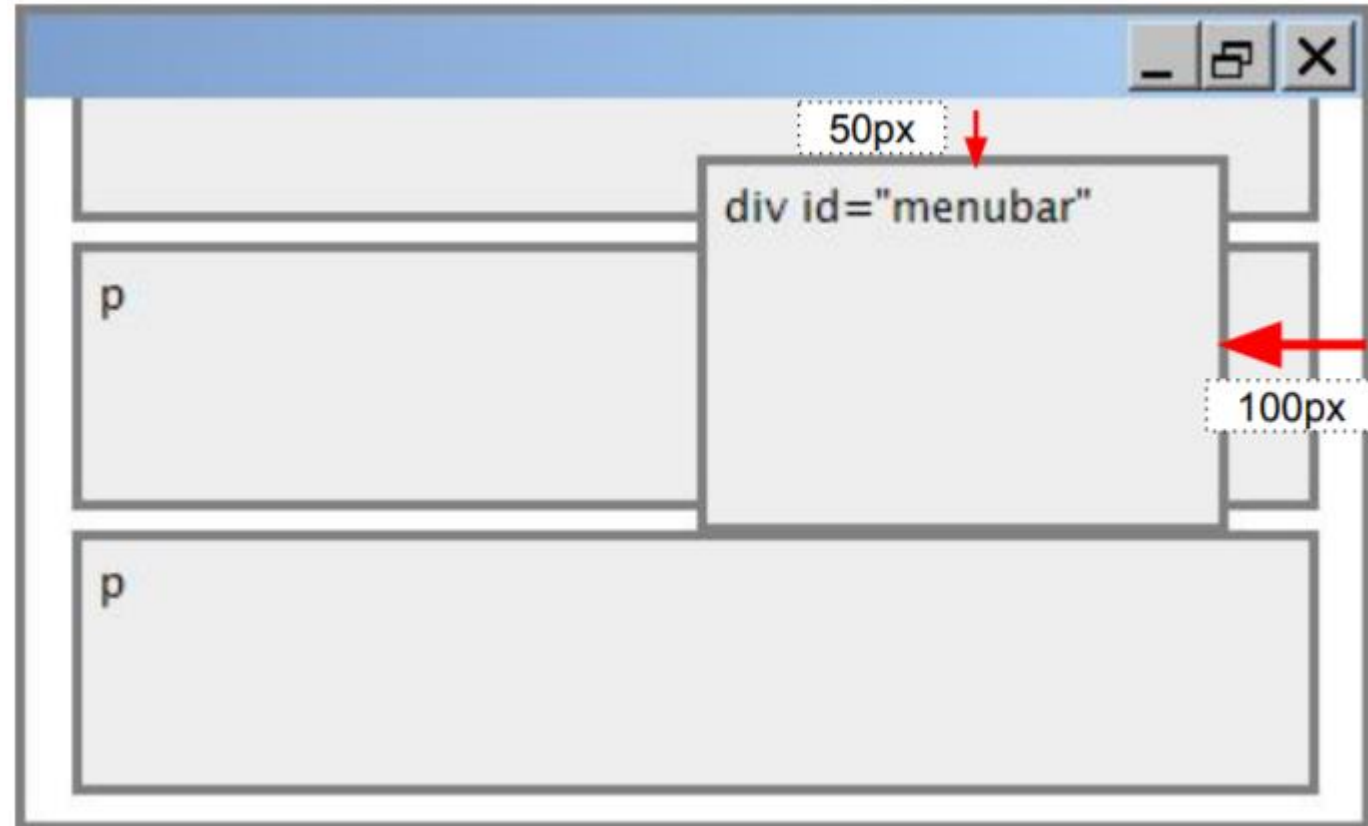
position: fixed

```css
#menubar {
    position: fixed;
    top: 50px;
    right: 100px;
    width: 400px;
}
```
CSS

- For **fixed positioning**, the offset is the distance positioned relative to the viewport.

- The element **does not move** when the webpage is scrolled.

- Element is removed from normal document flow, positioned on its own layer

Often used to implement UIs; control bars that shouldn't go away

position: fixed

```css
#box1 {                    CSS
  height: 50px;
  background-color:
    rgba(0, 0, 0, 0.5);
  position: fixed;
  top: 50%;
  left: 0;
  right: 0;
}
```

Codepen

vernacular English, puppy refers specifically to dogs, while pup may often be used for other mammals such as seals, giraffes, guinea pigs, or even rats.

Development

At first, puppies spend the large majority of their time sleeping and the rest feeding. They instinctively pile together into a heap, and become distressed if separated from physical contact with their littermates, by even a short distance.

Puppies are born with a fully functional sense of smell but can't open their eyes. During their first two weeks, a puppy's senses all develop rapidly. During this stage the nose is the primary sense organ used by puppies to find their mother's teats, and to locate their littermates, if they become separated by a short distance. Puppies open their eyes about nine to eleven days following birth. At first, their retinas are poorly developed and their vision is poor. Puppies are not able to see as well as adult dogs. In addition, puppies' ears remain sealed until about thirteen to seventeen days after birth, after which they respond more actively to sounds. Between two and four weeks old, puppies usually begin to growl, bite, wag their tails, and bark.

Puppies develop very quickly during their first three months, particularly after their eyes and ears open and they are no longer completely dependent on their mother. Their coordination and strength improve, they spar with their littermates, and begin to explore the world outside the nest. They play wrestling, chase, dominance, and tug-of-war games.

Development

Puppies are highly social animals and spend most of their waking hours interacting with either their mother or littermates. When puppies are socialized

position: absolute

```css
#menubar {
    position: absolute;
    left: 400px;
    top: 50px;
    width: 300px;
}
```
CSS

- For **absolute positioning**, the offset is the distance from the nearest **positioned** "ancestor element", if no positioned ancestor, document body is used
- Element is removed from normal document flow, positioned on its own layer
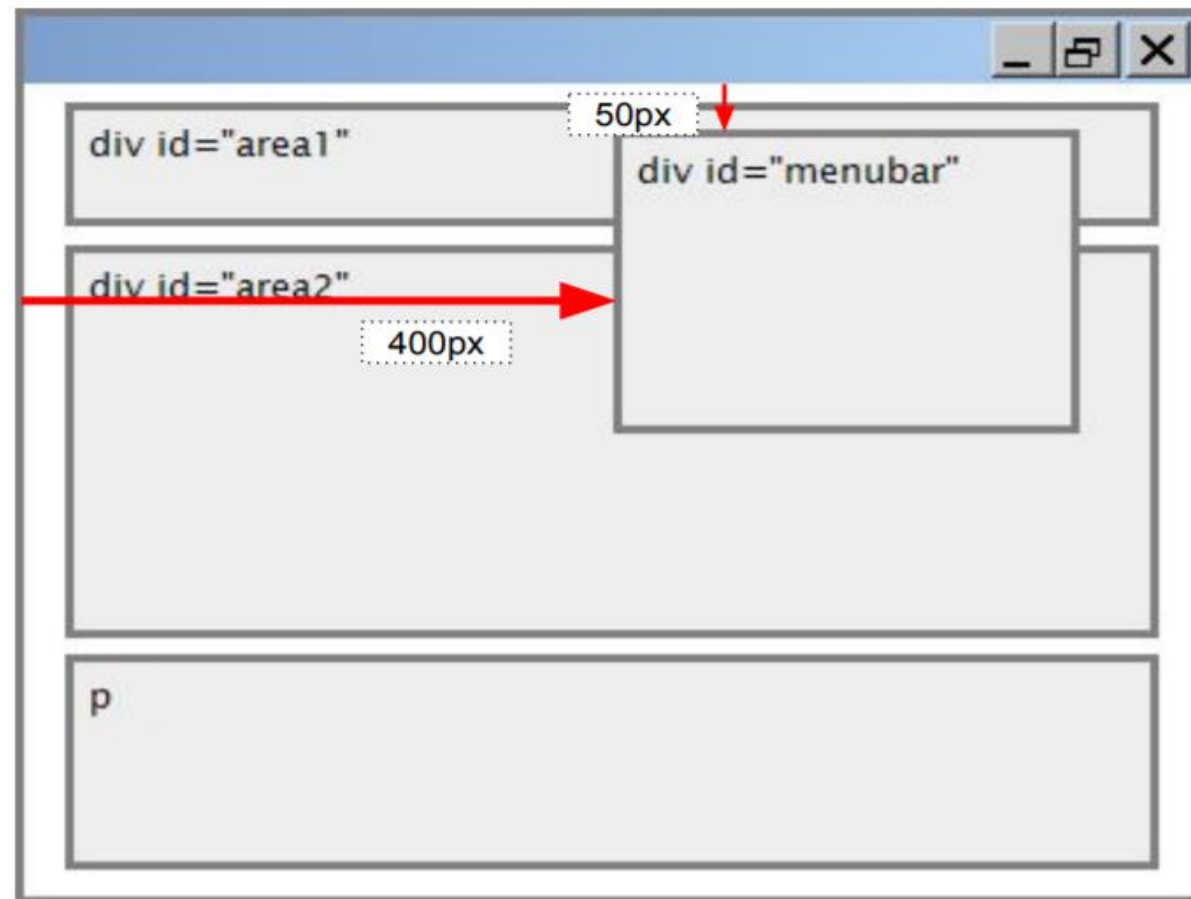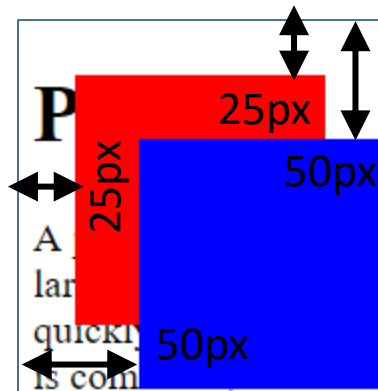
position: absolute

```css
#box1 {
  height: 100px;
  width: 100px;
  background-color: red;
  position: absolute;
  top: 25px;
  left: 25px;
}

#box2 {
  height: 100px;
  width: 100px;
  background-color: blue;
  position: absolute;
  top: 50px;
  left: 50px;
}
```

HTML

```html
<body>
  <h1>Puppy</h1>
  <p>A puppy is a juvenile d
1.36 kg), while larger ones ca
All healthy puppies grow quick
change as the puppy grows older
the Yorkshire Terrier. In verna
specifically to dogs, while pu
such as seals, giraffes, guine
  <h2>Development</h2>
  <p>At first, puppies spend
sleeping and the rest feeding.
heap, and become distressed if
their littermates, by even a s
  <div id="box1"></div>
  <div id="box2"></div>
</body>
```

25px

25px

50px

50px

P

A p ... dog. Some puppies can weigh 1–3 lb (0.45–1.36 kg), while
lar ... up to 15–23 lb (6.8–10.4 kg). All healthy puppies grow
quickl ... puppy's coat color may change as the puppy grows older, as
is com ... reeds such as the Yorkshire Terrier. In vernacular English,
puppy refers specifically to dogs, while pup may often be used for other mammals
such as seals, giraffes, guinea pigs, or even rats.

Development

At first, puppies spend the large majority of their time sleeping and the rest
feeding. They instinctively pile together into a heap, and become distressed if
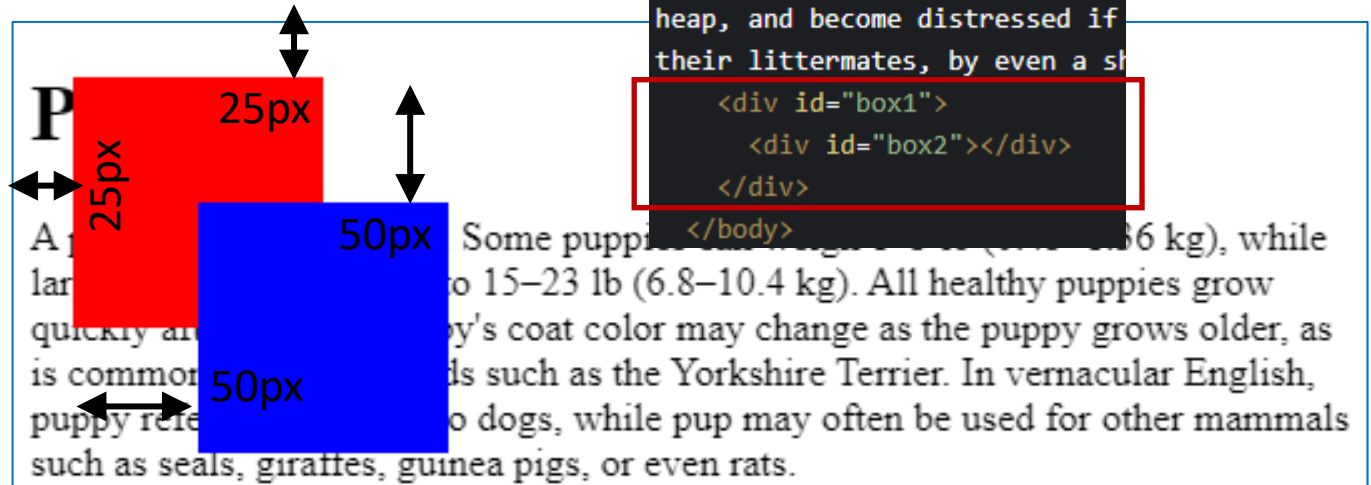separated from physical contact with their littermates, by even a short distance.

Codepen

position: absolute

```css
#box1 {
  height: 100px;
  width: 100px;
  background-color: red;
  position: absolute;
  top: 25px;
  left: 25px;
}

#box2 {
  height: 100px;
  width: 100px;
  background-color: blue;
  position: absolute;
  top: 50px;
  left: 50px;
}
```

HTML

```html
<body>
  <h1>Puppy</h1>
  <p>A puppy is a juvenile do
1.36 kg), while larger ones ca
All healthy puppies grow quick
change as the puppy grows olde
the Yorkshire Terrier. In vern
specifically to dogs, while pu
such as seals, giraffes, guine
  <h2>Development</h2>
  <p>At first, puppies spend
sleeping and the rest feeding.
heap, and become distressed if
their littermates, by even a s
  <div id="box1">
    <div id="box2"></div>
  </div>
  </body>
```

25px

25px

50px

50px

Puppy

A puppy is a juvenile d̶o̶g̶.̶ Some puppies̶ ̶c̶a̶n̶ ̶w̶e̶i̶g̶h̶ ̶1̶-̶3̶ ̶l̶b̶ ̶(̶0̶.̶4̶5̶-̶ ...36 kg), while larger ones can̶ ̶w̶e̶i̶g̶h̶ ̶u̶p̶ ̶t̶o̶ 15–23 lb (6.8–10.4 kg). All healthy puppies grow quickly a̶n̶d̶ ̶t̶h̶e̶ ̶p̶u̶p̶p̶y̶'s coat color may change as the puppy grows older, as is commo̶n̶ ̶i̶n̶ ̶b̶r̶e̶e̶d̶s such as the Yorkshire Terrier. In vernacular English, puppy refe̶r̶s̶ ̶o̶n̶l̶y̶ ̶t̶o̶ dogs, while pup may often be used for other mammals such as seals, giraffes, guinea pigs, or even rats.

Development

At first, puppies spend the large majority of their time sleeping and the rest feeding. They instinctively pile together into a heap, and become distressed if separated from physical contact with their littermates, by even a short distance.

Codepen

position: absolute

```
be used for other mammals such
or even rats.</p>
    <h2>Development</h2>
    <p>At first, puppies spend
sleeping and the rest feeding.
into a heap, and become distres
contact with their littermates
    <div id="outside">
        <div id="box1"></div>
        <div id="box2"></div>
    </div>
</body>
```
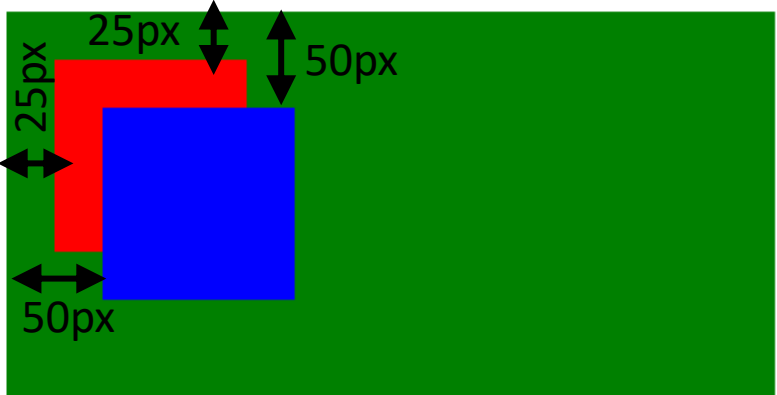
```css
#box1 {                              CSS
  height: 100px;
  width: 100px;
  background-color: red;
  position: absolute;
  top: 25px;
  left: 25px;
}

#box2 {
  height: 100px;
  width: 100px;
  background-color: blue;
  position: absolute;
  top: 50px;
  left: 50px;
}

#outside {
   position: absolute;
   background-color: green;
   width: 400px;
   height: 200px;
}
```
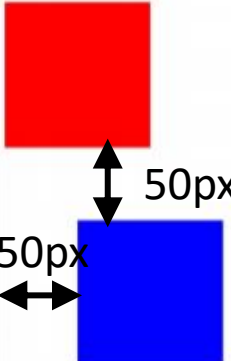
Codepen

Puppy

A puppy is a juvenile dog. Some puppies can weigh 1–3 lb (0.45–1.36 kg), while larger ones can weigh up to 15–23 lb (6.8–10.4 kg). All healthy puppies grow quickly after birth. A puppy's coat color may change as the puppy grows older, as is commonly seen in breeds such as the Yorkshire Terrier. In vernacular English, puppy refers specifically to dogs, while pup may often be used for other mammals such as seals, giraffes, guinea pigs, or even rats.

Development

At first, puppies spend the large majority of their time sleeping and the rest feeding. They instinctively pile together into a heap, and become distressed if separated from physical contact with their littermates, by even a short distance.

25px 25px 50px 50px

Without top/bottom, left/right offsets, defaults to the normal flow in document: as no position was applied

position: relative

- For `position: relative;` the element is placed where it would normally be placed in the layout of the page but shifted by the `top` / `left` / `bottom` / `right` values.

```css
#box2 {
    height: 100px;
    width: 100px;
    background-color: blue;
    position: relative;
    top: 50px;
    left: 50px;
}
```
CSS

[Codepen](#)

Puppy

A puppy is a juvenile dog. Some puppies can weigh 1–3 lb (0.45–1.36 kg), while larger ones can weigh up to 15–23 lb (6.8–10.4 kg). All healthy puppies grow quickly after birth. A puppy's coat color may change as the puppy grows older, as is commonly seen in breeds such as the Yorkshire Terrier. In vernacular English, puppy refers specifically to dogs, while pup may often be used for other mammals such as seals, giraffes, guinea pigs, or even rats.

Development

At first, puppies spend the large majority of their time sleeping and the rest feeding. They instinctively pile together into a heap, and become distressed if separated from physical contact with their littermates, by even a short distance.

50px

50px

Positioning elements with `float`

- The float property specifies how an element should float:
  - left - The element floats to the left of its container
  - right- The element floats to the right of its container
  - none - The element does not float (will be displayed just where it occurs in the text). This is default
  - inherit - The element inherits the float value of its parent
- Note: Absolutely positioned elements ignores the float property!
- Note: **Elements after a floating element will flow (wrap) around it**
  - to avoid this, use the clear property or the clearfix hack

Positioning elements with `float`
(example without float)

- `<div>` takes the height of the containing element(s): **`<p> <img>`**

```html
<h2>UCY campus</h2>
<div>
  <img src="ucy-cafe.jpg"/>
  <p>Welcome to the UCY campus</p>
</div>
```

```css
div {
    background-color: lightblue;
}
img {
    height: 200px;
}
```

UCY campus

Welcome to the UCY campus

Both `<p>` and `<img>` are contained within the `<div>` so the lightblue background covers both of them.

Positioning elements with `float`

(example **with** float)

- `<div>` takes the height of the containing element(s): **`<p>`**

```html
<h2>UCY campus</h2>                          HTML
<div>
  <img src="ucy-cafe.jpg"/>
  <p>Welcome to the UCY campus</p>
</div>
```

```css
div {                                        CSS
    background-color: lightblue;
}
img {
    height: 200px;
    float: left;
}
```

UCY campus

Welcome to the UCY campus   `<div>`

Floated element `<img>` is not contained within the `<div>` anymore (actually `<img>` is removed from the normal document flow, and floats to the left of `<div>`) so the lightblue background covers only the `<p>` which is still contained within the `<div>`

Positioning elements with `float`

(example **with** float)

- `<div>` takes the height of the containing element(s): **`<p>`**

```html
<h2>UCY campus</h2>
<div>
  <img src="ucy-cafe.jpg"/>
  <p>Welcome to the UCY campus</p>
</div>
```

```css
div {
    background-color: lightblue;
    overflow: hidden;
}
img {
    height: 200px;
    float: left;
}
```

UCY campus

Welcome to the UCY campus

`overflow:hidden` hides any part of an element which flow out of the parent container. At the same time, it "forces" the child elements to be contained within the parent element.

Positioning elements with `display:flex`

```html
<div class="parent">
  <div class="child">
    <h2>First paragraph</h2>
    Lorem ipsum dolor sit amet, consectetur
adipiscing elit, ...
  </div>
  <div class="child">
    <h2>Second paragraph</h2>
    Lorem ipsum dolor sit amet, consectetur
adipiscing elit, ...
  </div>
  <div class="child">
    <h2>Third paragraph</h2>
    Lorem ipsum dolor sit amet, consectetur
adipiscing elit, ...
  </div>
  <div class="child">
    <h2>Fourth paragraph</h2>
    Lorem ipsum dolor sit amet, consectetur
adipiscing elit, ...
  </div>
</div>
```
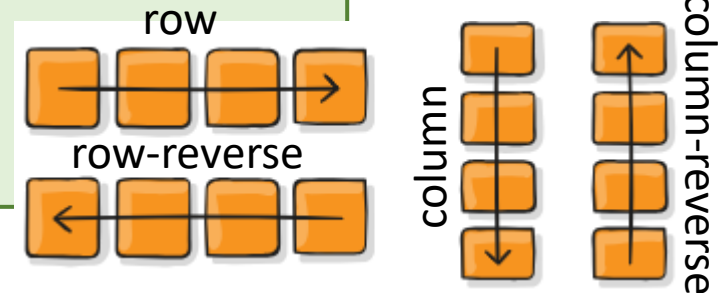
CSS

```css
.parent {
  display: flex;
  flex-flow: row wrap;
}

.child {
  width: 40%;
}
```

row

row-reverse

column

column-reverse

First paragraph

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Second paragraph

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Third paragraph

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Fourth paragraph

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

https://codepen.io/paulantoniou/pen/ExMqmom

Positioning elements with `display:flex`

```html
<div class="parent">
  <div class="child">
    <h2>First paragraph</h2>
    Lorem ipsum dolor sit amet, consectetur
adipiscing elit, ...
  </div>
  <div class="child">
    <h2>Second paragraph</h2>
    Lorem ipsum dolor sit amet, consectetur
adipiscing elit, ...
  </div>
  <div class="child">
    <h2>Third paragraph</h2>
    Lorem ipsum dolor sit amet, consectetur
adipiscing elit, ...
  </div>
  <div class="child">
    <h2>Fourth paragraph</h2>
    Lorem ipsum dolor sit amet, consectetur
adipiscing elit, ...
  </div>
</div>
```

CSS

```css
.parent {
  display: flex;
  flex-flow: row nowrap;
}


.child {
  width: 40%;
}
```

row

row-reverse

column

column-reverse

First paragraph

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Second paragraph

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Third paragraph

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Fourth paragraph

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
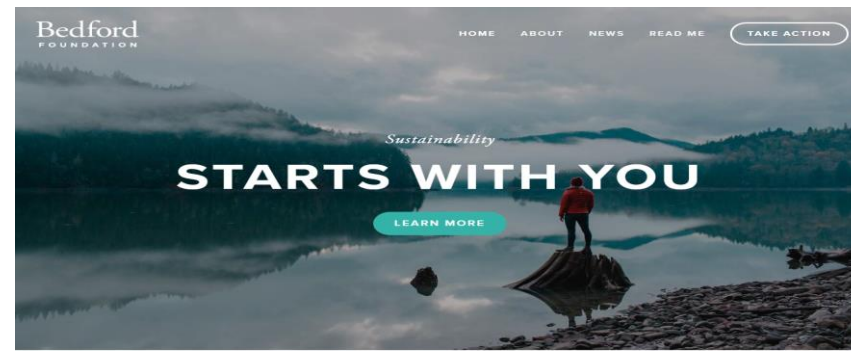
https://codepen.io/paulantoniou/pen/ExMqmom

Positioning elements with `display:flex`

```html
<div class="parent">
  <div class="child">
    <h2>First paragraph</h2>
    Lorem ipsum dolor sit amet, consectetur
adipiscing elit, ...
  </div>
  <div class="child">
    <h2>Second paragraph</h2>
    Lorem ipsum dolor sit amet, consectetur
adipiscing elit, ...
  </div>
  <div class="child">
    <h2>Third paragraph</h2>
    Lorem ipsum dolor sit amet, consectetur
adipiscing elit, ...
  </div>
  <div class="child">
    <h2>Fourth paragraph</h2>
    Lorem ipsum dolor sit amet, consectetur
adipiscing elit, ...
  </div>
</div>
```

```css
.parent {
  display: flex;
  flex-flow: row wrap;
  justify-content: center;
}


.child {
  width: 40%;
}
```


flex-start

flex-end

center

space-between

space-around

space-evenly

Defines the alignment along the main axis. It helps distribute extra free space leftover when either all the flex items on a line are inflexible, or are flexible but have reached their maximum size.

https://codepen.io/paulantoniou/pen/ExMqmom

Positioning elements with `display:flex`

HTML

```html
<div class="parent">
  <div class="child">
    <h2>First paragraph</h2>
    Lorem ipsum dolor sit amet, consectetur
adipiscing elit, ...
  </div>
  <div class="child">
    <h2>Second paragraph</h2>
    Lorem ipsum dolor sit amet, consectetur
adipiscing elit, ...
  </div>
  <div class="child">
    <h2>Third paragraph</h2>
    Lorem ipsum dolor sit amet, consectetur
adipiscing elit, ...
  </div>
  <div class="child">
    <h2>Fourth paragraph</h2>
    Lorem ipsum dolor sit amet, consectetur
adipiscing elit, ...
  </div>
</div>
```

CSS

```css
.parent {
  display: flex;
  flex-flow: row wrap;
  justify-content: center;
  gap: 10px 40px;
}


.child {
  width: 40%;
}
```

gap: 10px

gap: 30px
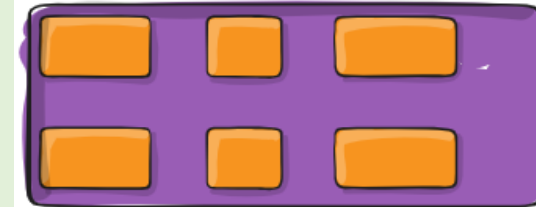
gap: 10px 30px

First paragraph

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Second paragraph

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Third paragraph

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Fourth paragraph

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

https://codepen.io/paulantoniou/pen/ExMqmom

Layout Exercise

- Squarespace's most popular template looks like this:

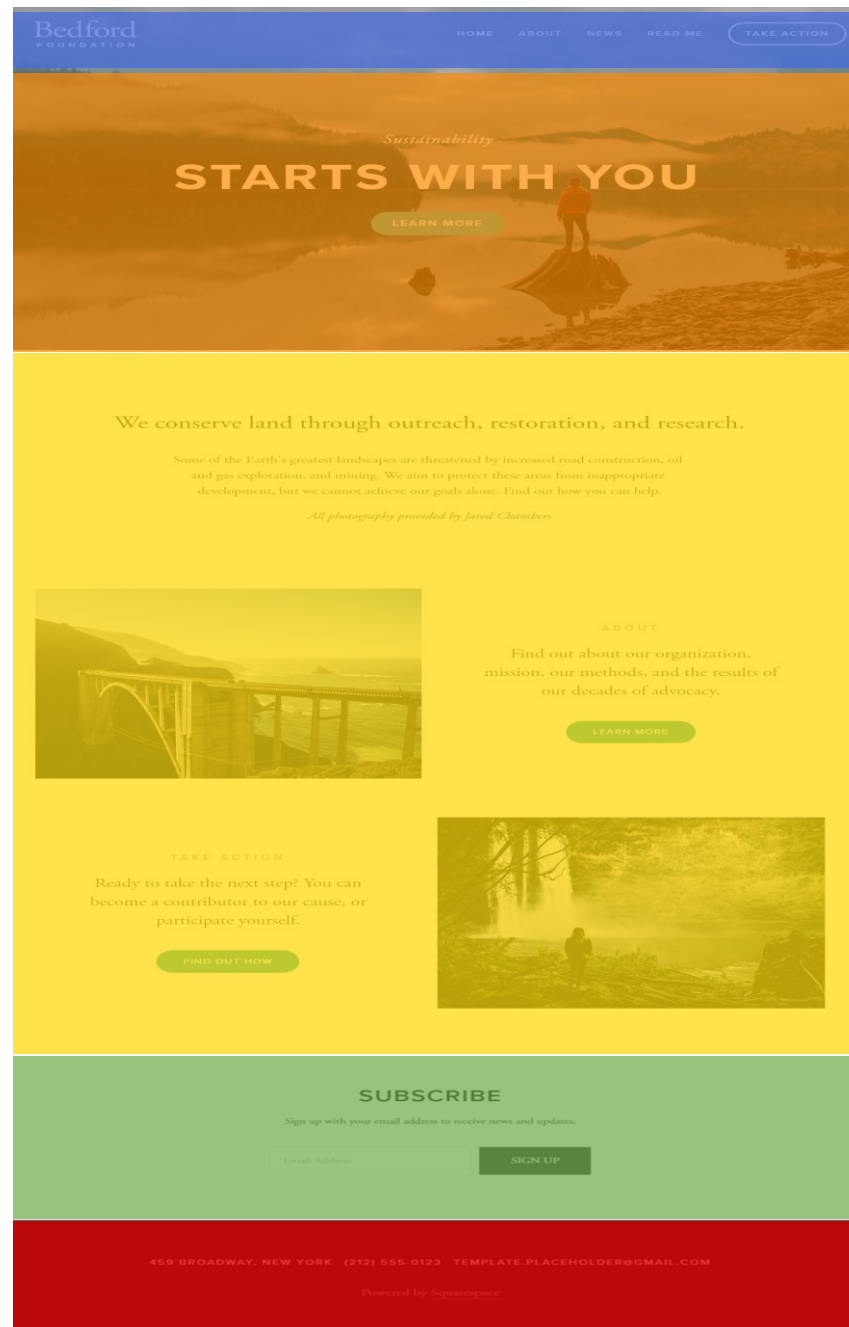Q: Do we know enough to make something like that?

Basic shape

- Begin visualizing the layout in terms of boxes:

Basic shape

- Begin visualizing the layout in terms of boxes:

- **Let's first try making this layout!**

Content Sectioning elements

These elements do not "do" anything; they are basically more descriptive `<div>`s. Makes your HTML more readable. See MDN for more info.

Name	Description
`<p>`	Paragraph (mdn)
`<h1>`-`<h6>`	Section headings (mdn)
`<article>`	A document, page, or site (mdn) This is usually a root container element after body
`<section>`	Generic section of a document (mdn)
`<header>`	Introductory section of a document (mdn)
`<footer>`	Footer at end of a document or section (mdn)
`<nav>`	Navigational (menu) section (mdn)
`<aside>`	Sidebar (mdn)

Content Sectioning elements

- Prefer these elements to <div>s when it makes sense!
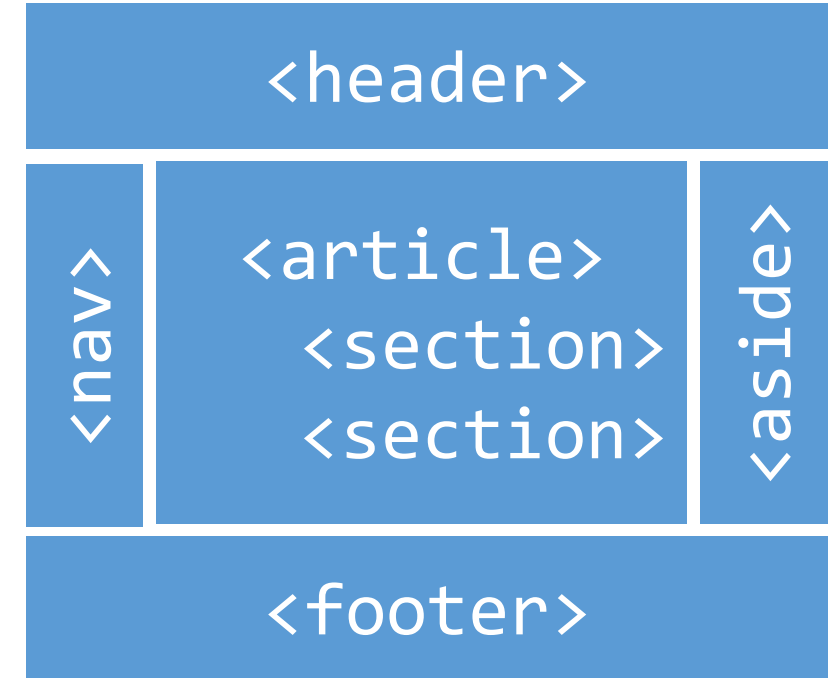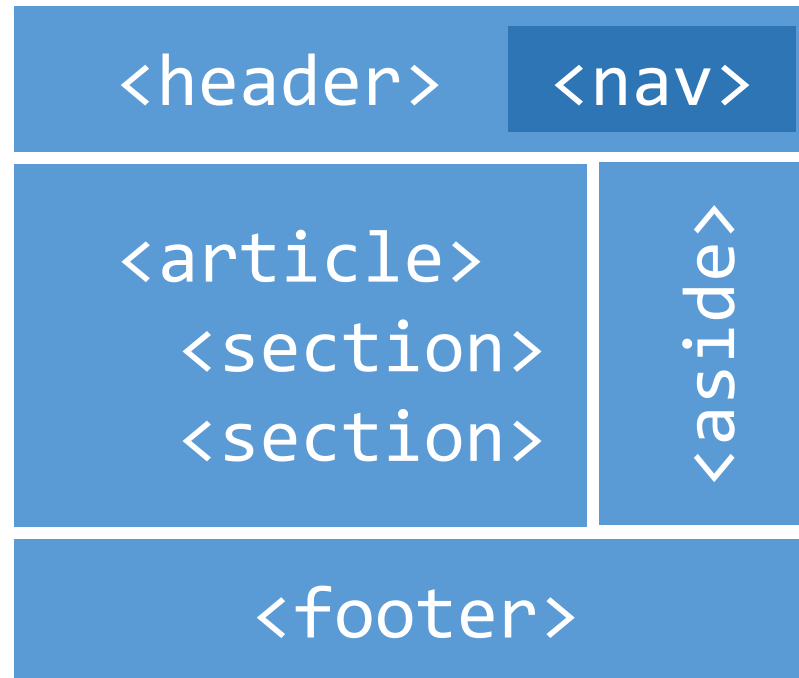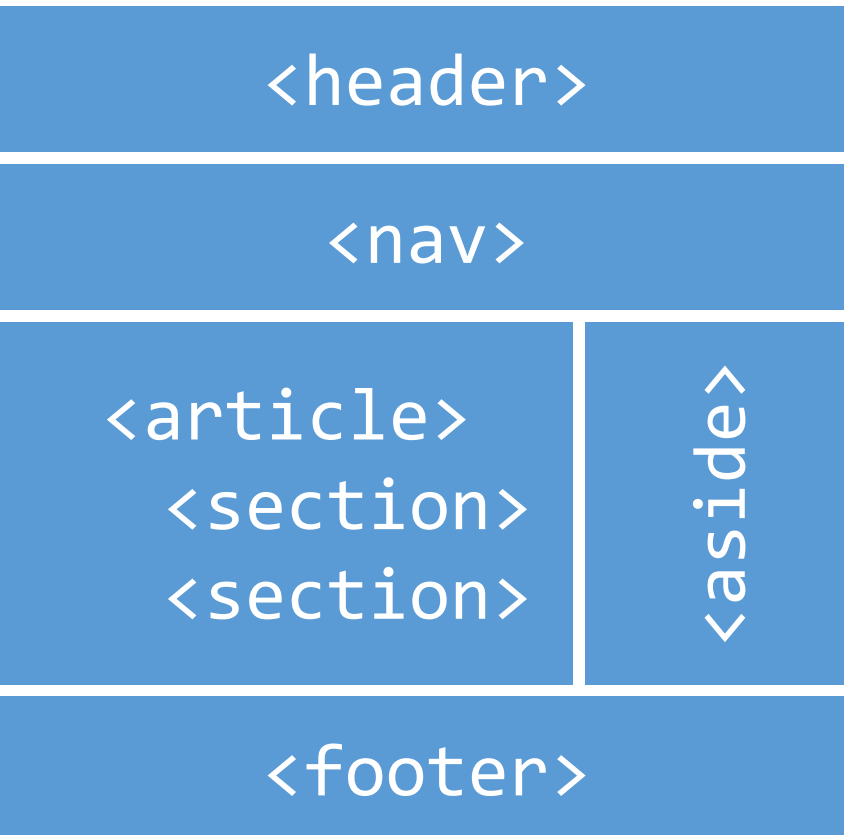
Top

- Navbar:
  - Height: 75px
  - Background: royalblue
  - `<nav>`
- Header:
  - Height: 400px;
  - Background: darkorange
  - `<header>`
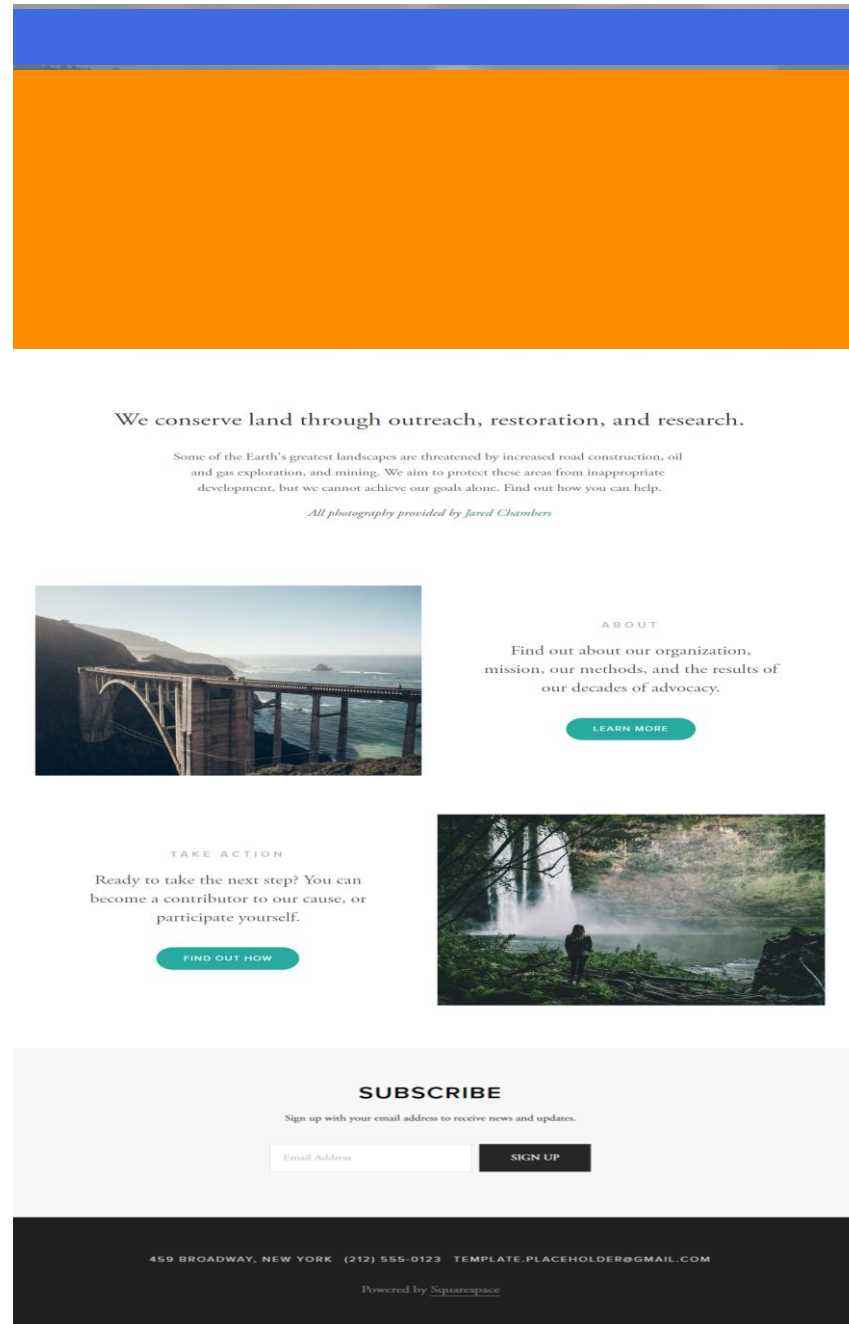
Main

- Gold box:
  - Surrounding space: 96px above and below; 32px on each side
  - Height: 990px
  - Background: gold
  - `<section>` ή `<article>`
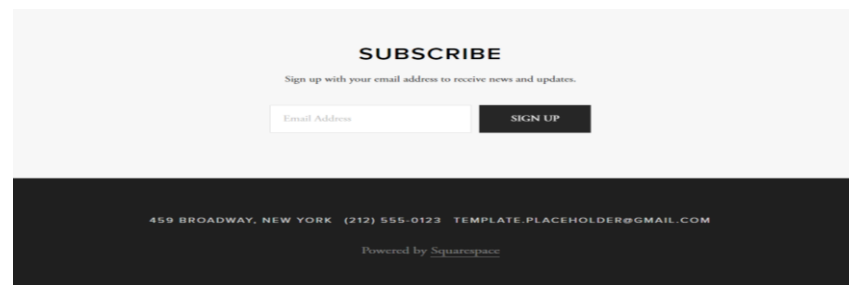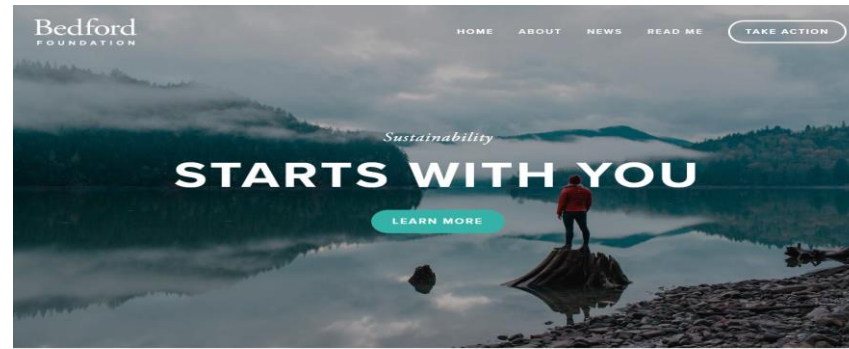
Bottom

- Subscribe:
  - Height: 310px
  - Background: darkgray
  - `<section>`

- Footer
  - Height: 210px
  - Background: black
  - `<footer>`
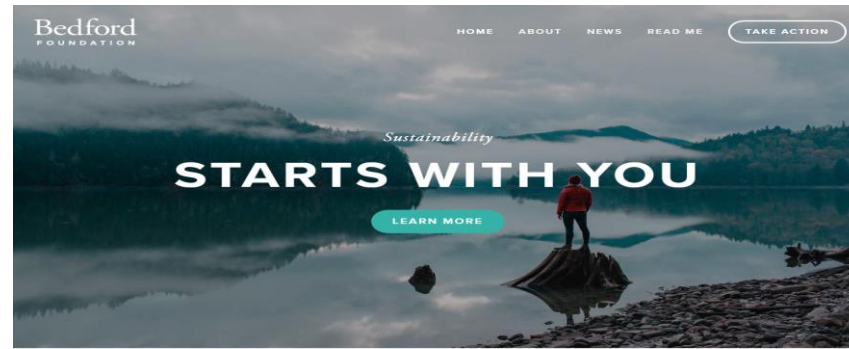
Main contents

- Title with paragraphs:
  - Height: 290px
  - Background: khaki
  - Space beneath: 80px
  - `<h1>`, `<p>` within `<div>`

- Images and Descriptions
  - Height: 275px;
  - Width: 48% of the parent's width, with space in between
  - Horizontal space: 80px;
  - Background: forestgreen
  - `<div>`
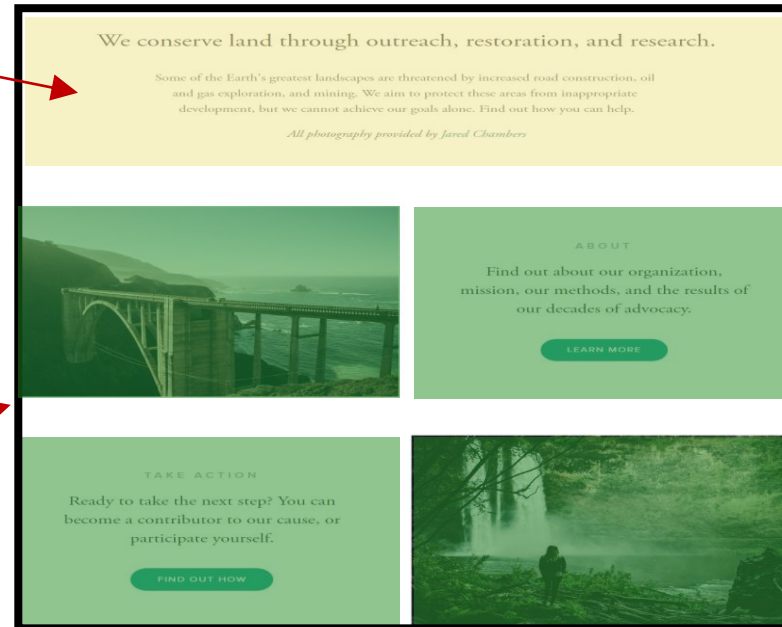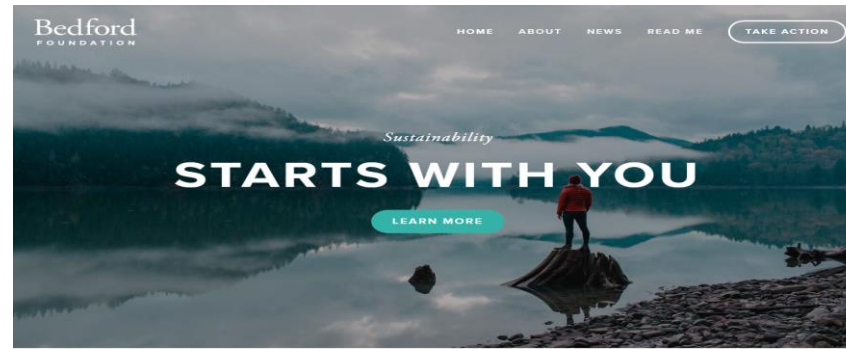
Codepen

Can be developed using either `position` property (see Codepen above), or with `float` property.
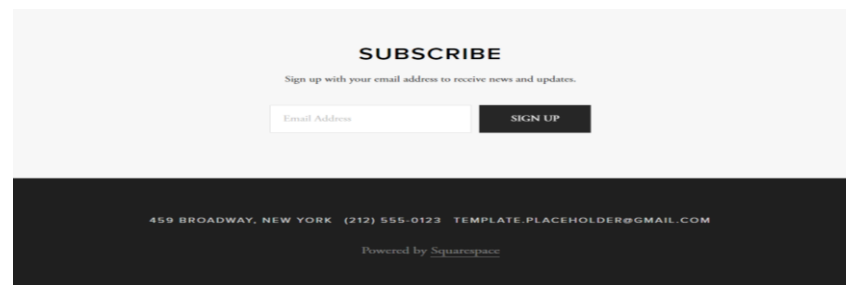
Random useful CSS

How to set an image within a div

- 1st way: Use <img> inside <div>

```
<div class="header">
    <img src=background.png" height="400px"/>
</div>
```

- 2nd way: Set it as background-image on <div> element

```
<div class="header">
</div>
```

```
.header {
    background-image: url(background.png);
    height: 400px;
}
```

background properties

- You can then use [additional background properties](#) to further style it:

```
background-size: cover;
background-size: contain;
background-repeat: no-repeat;
background-position: top;
background-position: center;
background-attachment: fixed;
```

- ([CodePen](#): Try resizing the window)

background-repeat property

```css
.header {                                          CSS
    background-image: url(https://cs.ucy.ac.cy/images/ucy_logo_en.png);
    height: 195px;
}
```



```css
.header {                                          CSS
    background-image: url(https://cs.ucy.ac.cy/images/ucy_logo_en.png);
    background-repeat: no-repeat;
    height: 195px;
}
```

background-size property

```css
.header {                                          CSS
    background-image: url(https://cs.ucy.ac.cy/images/ucy_logo_en.png);
    height: 195px;
    background-size: contain;
}
```


contain: Scales the image as large as possible (e.g. here height to be 195px) <u>without cropping or distorting</u> the image (show the whole image). If space remains, another image is inserted.

`background-size` property

```
header {
    background-image: url(https://cs.ucy.ac.cy/images/ucy_logo_en.png);
    height: 195px;
    background-size: cover;
}
```

cover: Scales the image as large as possible <u>without distorting</u> the image to cover width and height. If the proportions of the image differ from the element, it is <u>cropped either vertically or horizontally so that no empty space remains</u>. Above image is stretched to cover whole width, even that is cropped below.

`background-attachment` property

- Parallax scrolling is a web site trend where the background content (i.e. an image) is moved at a different speed than the foreground content while scrolling:
  - create a container element (e.g. a div)
  - add a background image to the container with a specific height
  - add background-attachment: fixed to create the actual parallax effect
  - other background properties can be used to center and scale the image perfectly

- See examples here:
  - W3 School: Parallax demo, No parallax
  - Codepen

`background-image`: Gradients

- CSS gradients let you display smooth transitions between two or more specified colors

- CSS defines two types of gradients:

    o Linear gradients (down/up/left/right/diagonally)

    ```
    background-image: linear-gradient(direction, color-stop1, color-stop2, ...);
    ```

    o Radial gradients (defined by the center)

    ```
    background-image: linear-gradient(shape size at position, start-color, ...,
    last-color);
    ```

background-image: Gradients

```css
#grad1 {                                       CSS
  width: 200px;
  height: 150px;
   /* For browsers that do not support gradients */
  background-color: red;
   /* Standard syntax (must be last) */
  background-image: linear-gradient(red, yellow);
}
```

```css
#grad1 {                                       CSS
  width: 200px;
  height: 150px;
   /* For browsers that do not support gradients */
  background-color: red;
   /* Standard syntax (must be last) */
  background-image: radial-gradient(circle, red,
yellow, green);
}
```

`background-image`: Gradients

- More tutorials: https://www.w3schools.com/css/css3_gradients.asp

- Some gradient generators: https://mycolor.space/gradient, https://www.css-gradient.com/, https://webgradients.com/

Web Fonts

- You can use [Google Fonts](#) to choose from a set of visually appealing fonts:

Web Fonts

- Select font family (from previous slide)

- Click "Get Font"

- Click on the "Get embed code"
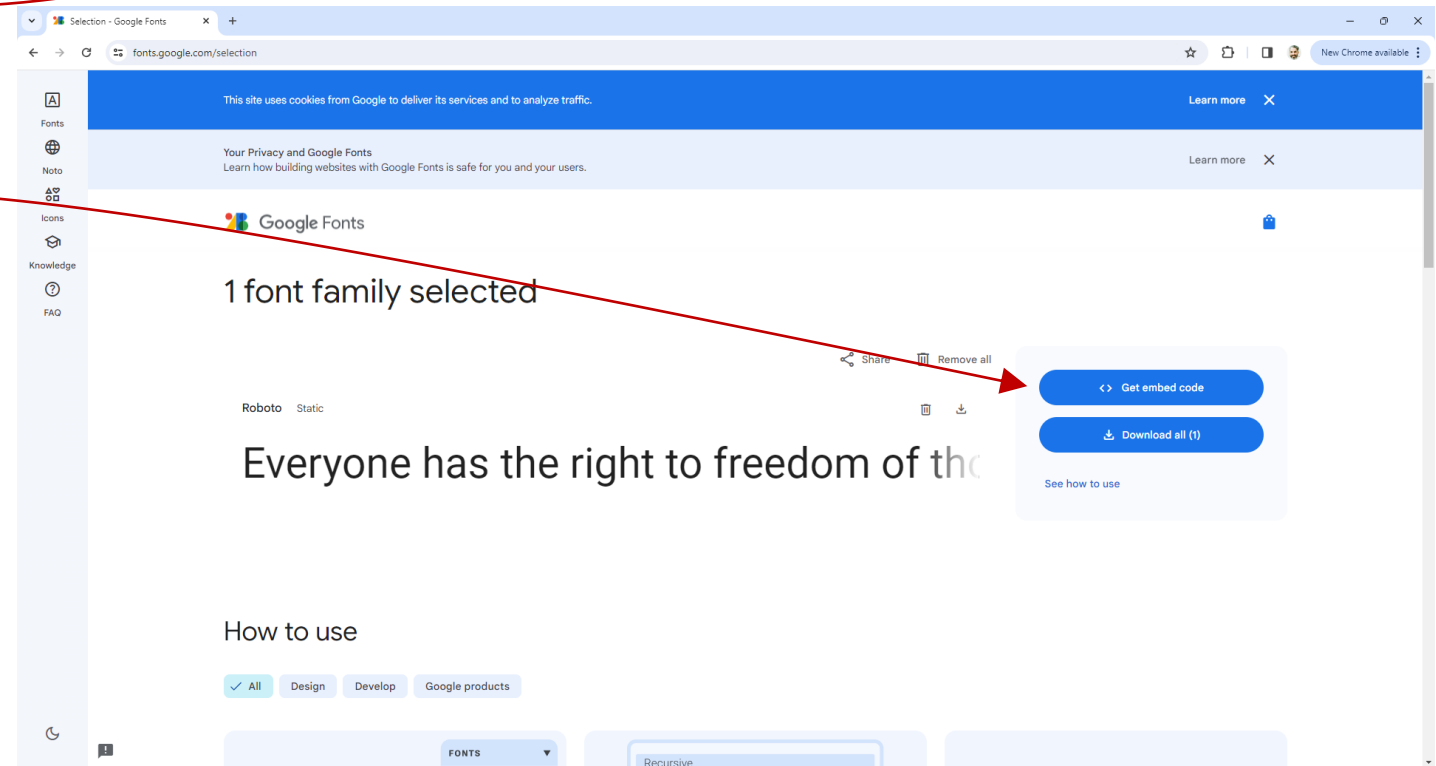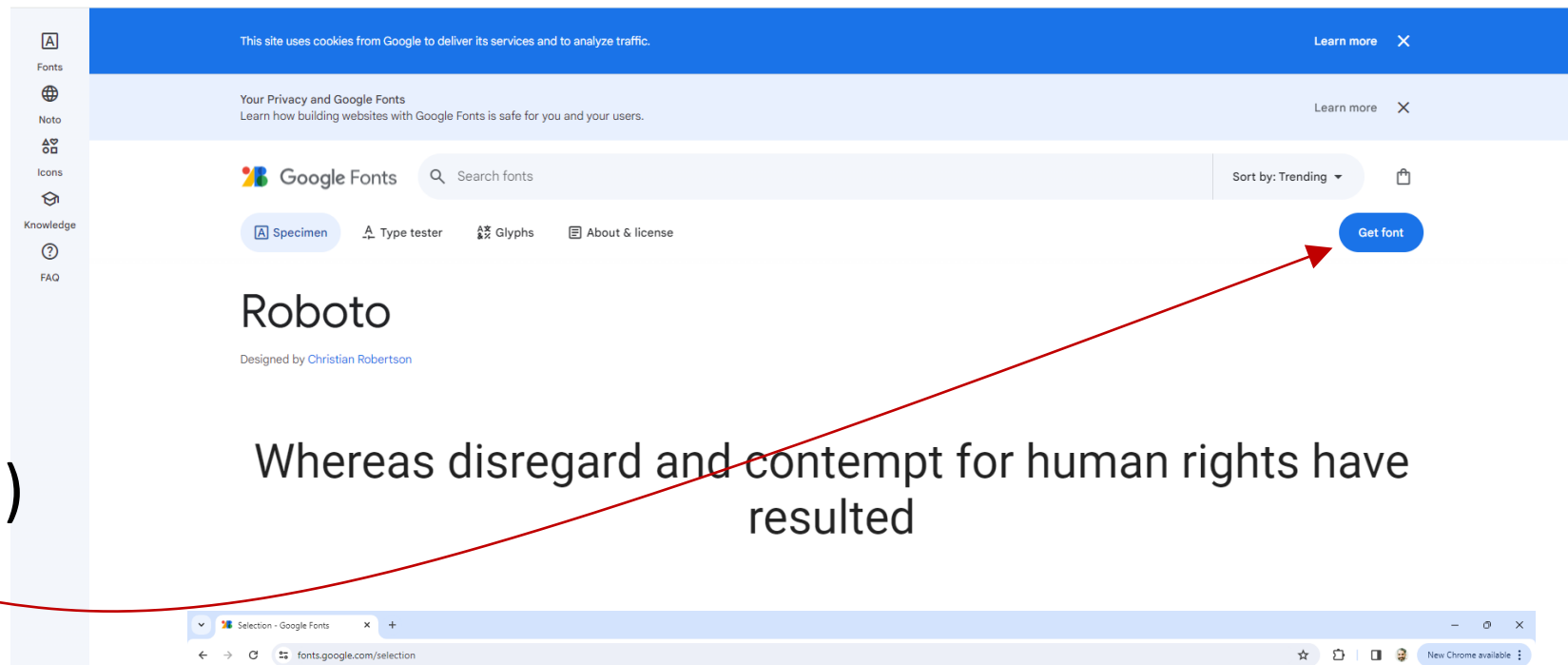
Web Fonts

- Copy the given `<link>` tag into the `<head>` section of your page alongside your other CSS files

Aside: Fallback fonts

- Notice that the Google Font example shows a comma-separated list of values for `font-family`:

```
font-family: 'Roboto', sans-serif;
```

- Each successive font listed is a fallback, i.e. the font that will be loaded if the previous font could not be loaded

- There are also six [generic font names](#) which allows the browser to choose the font based on intent + fonts available on the OS.

- It's good practice to list a generic font at the end of all your `font-family` declarations.

overflow property

- Sets what to do when a content is too big to fit in its element's box

overflow: visible

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

```
div.ex4 {                            CSS
    background-color: lightblue;
    width: 110px;
    height: 110px;
    overflow: visible;
}
```

overflow: hidden ;

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh

overflow: scroll ; overflow: auto ;

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy

- Auto only shows a scrollbar when any content is clipped
- Scroll always shows the scrollbar even if all content fits and you can't scroll it (visible but disabled).

- The overflow property **works only for block** elements with a specified height.

W3School Example

overflow property

```
<div class="item">
   <img src="cs.jpg"/>
</div>
<div class="item">In this example
the image will float to the left.
</div>
```

```
.item {
    float: left;
    width: 300px;
    color: cyan;
    font-size: 22px;
}
```


300px

In this example the image will float to the left.

overflow property

```html
<div class="item">
   <img src="cs.jpg"/>
</div>
<div class="item">In this example
the image will float to the left.
</div>
```
HTML

```css
.item {
   float: left;
   width: 300px;
   color: cyan;
   font-size: 22px;
   overflow: hidden;
}
```
CSS

300px

In this example the image will float to the left.

Shadow Effects

- `text-shadow` property

```css
h1 {                                    CSS
    /* [x-offset] [y-offset] [blur] [color] */
    text-shadow: 2px 2px 5px red;
}
```

Text shadow effect!

- `box-shadow` property

```css
div {                                   CSS
    /* [x-offset] [y-offset] [blur] [color] */
    box-shadow: 10px 10px 5px grey;
}
```

This is a div element with a box shadow effect!
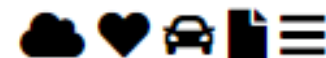
CSS Icons – Font Awesome icons

- To use Font Awesome icons, add the following line inside the <head> section of your HTML webpage:

```
<link rel="stylesheet"
href="https://use.fontawesome.com/
releases/v5.6.3/css/all.css">
```

```
<p>Some Font Awesome icons:</p>          HTML
<i class="fa fa-cloud"></i>
<i class="fa fa-heart"></i>
<i class="fa fa-car"></i>
<i class="fa fa-file"></i>
<i class="fa fa-bars"></i>
```

Some Font Awesome icons:

CSS Icons – Google Material icons

- To use Google Material icons, add the following line inside the <head> section of your HTML webpage:

```
<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
```

```
<p>Some Google icons:</p>
<i class="material-icons">cloud</i>
<i class="material-icons">favorite</i>
<i class="material-icons">attachment</i>
<i class="material-icons">computer</i>
<i class="material-icons">traffic</i>
```
HTML

Some Google icons:

Simplicity above all else

- Always prefer simplicity.
- Other tips:
  - **Separation of concerns**: HTML should contain content NOT style, CSS should contain style NOT content
  - **Descriptive HTML tags**: Make your HTML more readable by using `<header>` e.g. instead of `<div>` when appropriate
  - **Reduce redundancy**: Try grouping styles, using descendant selectors to reduce redundancy (see past slides for details)