

LAYOUTS

Using CSS to lay out your page

Lecture 3: Jan. 12 2016

Floats & Positioning

CSS layout properties

Normal Flow

`<p>...</p>`

`<p>...</p>`

`<p>...</p>`

This is the default behavior.

Every block-level element appears on a new line, causing each item to appear lower down the page than the previous one. Inline elements appear adjacent to each other.

Float

left | right | none | inherit

<p>...</p>

```

```

```

<p>...</p>
```

```
img.example {
  float: right;
  margin: 0 0 .25em .25em;
}
```

A float is a box that is shifted to the left or right on the current line. The most interesting characteristic of a float (or “floated” or “floating” box) is that content may flow along its side. Content flows down the right side of a left-floated box and down the left side of a right-floated box.

```
<div  
class="  
column"  
>...  
</div>
```

```
<div class="  
column">...  
</div>
```

```
<div  
class=  
"  
column  
">...  
</div>
```

- Images
- Drop caps
- Column / Grid layouts

```
.column {  
    float: left;  
}
```

Clear

left | right | both | none | inherit

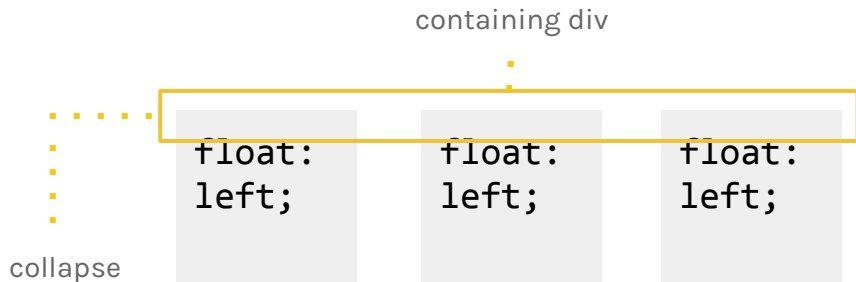


An element that has the clear property set on it will not move up adjacent to the float like the float desires, but will move itself down past the float.

An element with `clear: left;` will sit below a previous element that has a `float: left;` property applied. Same `clear: right;`.

An element with `clear: both;` will sit below elements with either `float: left;` or `float: right;`

Collapsed parents



```
<div class="parent">
  <div class="floatedchild">...</div>
  <div class="floatedchild">...</div>
  <div class="floatedchild">...</div>
</div>
```

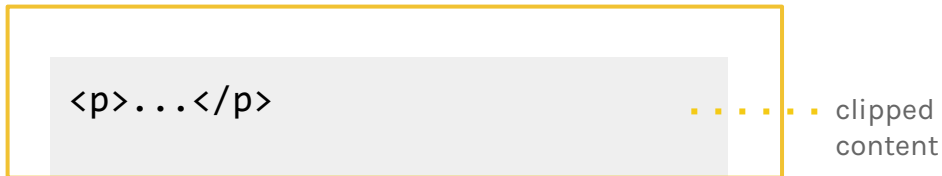
Specifying floats on child elements can affect their containing parent. If this parent element contained nothing but floated elements, the height of it would collapse.

The float must be cleared before the parent element is closed. There are a few ways to do this, but a common “hack” is to use `overflow: hidden;` or `overflow: auto;` on the parent element. Note: overflow property is **not** intended for float clearing.

Overflow

visible | hidden | scroll | auto | inherit

container with fixed height set with `overflow: hidden;`



Overflow accounts for when your content doesn't fit within the container and specifies how the text overflows.

Positioning

`static | relative | absolute | fixed`

Positioning allows you to control the layout of a page using the `position` property.

While floats are useful for created elements that interact with / flow with the rest of the document, positioning allows you to establish independent elements.

Relative Positioning

```
<p>...</p>
```

```
<p class="example">...</p>
```

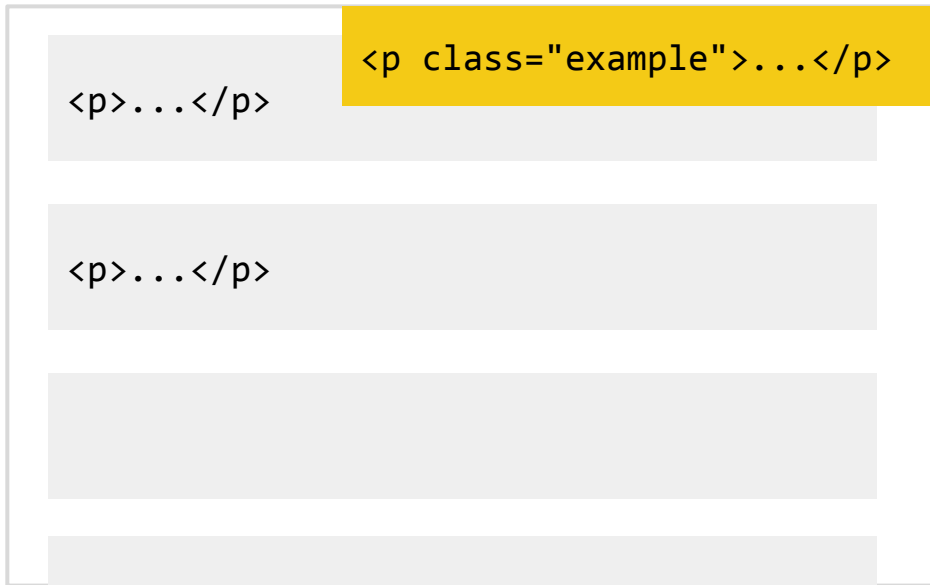
```
<p>...</p>
```

```
p.example {  
  position: relative;  
  top: 30px;  
  left: 100px;}
```

This offsets an element from the position it would be in **normal flow**, shifting it to the top, right, bottom, or left of where it would have been placed. This does not affect the position of surrounding elements; they stay in the position they would be in in normal flow.

```
<p>...</p>  
<p class="example">...</p>  
<p>...</p>
```

Absolute Positioning



```
p.example {  
  top: 0;  
  right: 0;  
  width: 60%;  
}
```

The element's box is completely removed from the flow of the document and positioned with respect to its **containing block**, which may be another element in the document or the initial containing block. Whatever space the element might have occupied in the normal document flow is closed up, as though the element did not exist.

```
<p>...</p>  
<p class="example">...</p>  
<p>...</p>
```

The Containing Block

Below are some nuances on positioning. The containing block is your positioning context or reference point.

- Elements can be positioned outside of their containing block.
- For **relatively positioned elements**, the containing block is formed by the **content edge** of the nearest block-level, tablecell, or inline-block parent. The reference point is the position the block would take if it were not moved.
- For **absolutely positioned** elements the containing block is set to the nearest **ancestor** (of any kind) that has a position value **other than static** (default).
 - Note: ancestor, not necessarily parent (since parent may have static positioning)
 - If the ancestor is block-level, the containing block is set to that **element's padding edge**; in other words, the area that would be bounded by a border.
 - If the ancestor is inline-level, the containing block is set to the **content edge** of the ancestor.

Fixed Positioning

```
<p class="example">...</p>
```

```
<p>...</p>
```

```
<p>...</p>
```

```
<p>...</p>  
<p class="example">...</p>  
<p>...</p>
```

```
p.example {  
  top: 0;  
  left: 0;  
}
```

The element's box behaves as though it were set to absolute, but its containing block is the **viewport**.

Viewport = boundaries of browser window

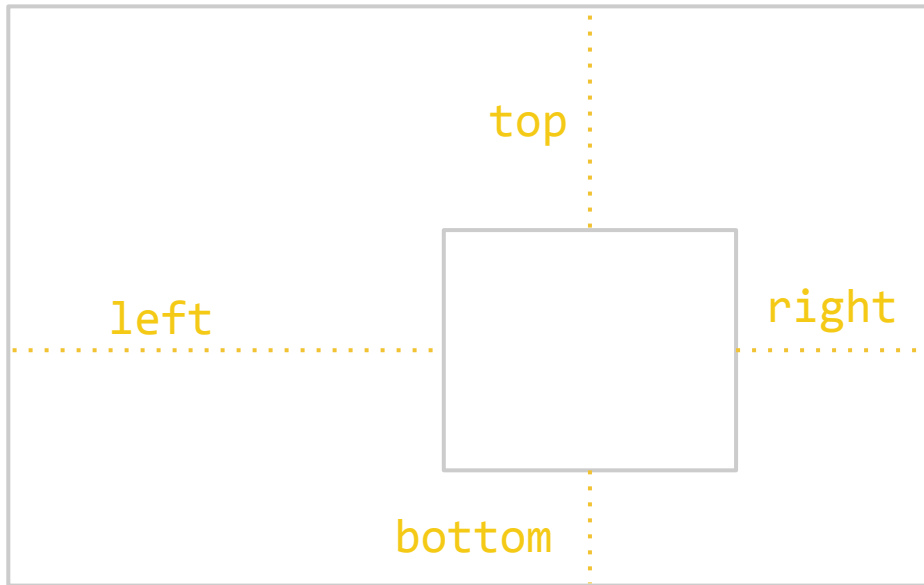
This means that fixed position elements don't move if the page is scrolled.

Offset Properties

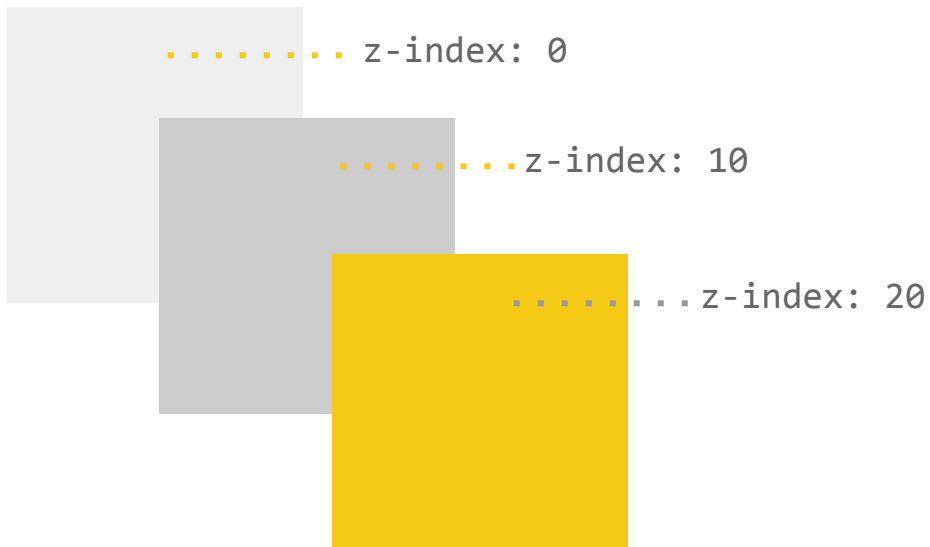
`top` | `right` | `bottom` | `left`

These properties describe an offset from the nearest side of the containing block. Think of them as describing a coordinate system.

Units: any length unit (px, em, %)



Z-index



The z-index value specifies the stack order of an element. This is similar to how layers work in Photoshop.

An element with a bigger number is always in front of an element with a smaller number.

The default is 0. It is good to use intervals of 10 in case additional elements need to be placed in between.

Hierarchy and Grids

Layout structures

Hierarchy

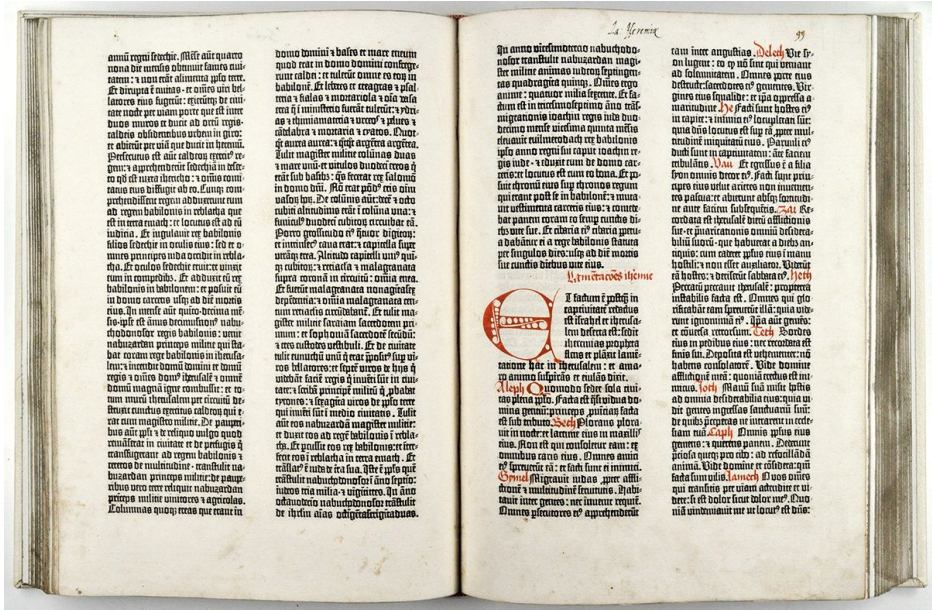
Visual hierarchy is the order in which the eye perceives what it sees. This order is created by visual contrast between forms in a perceptual field.

Content and tasks need to be grouped and sorted into a hierarchy to guide the viewer through their experience. Designers establish hierarchy by making adjustments to:

- Typography
- Color
- White Space
- Proximity
- Repetition
- Alignment
- Images
- Rules
- Grid

Visual hierarchy can inform our understanding of the goals of the website. Design decisions reflects what and how important are the site features with which you think the we should engage. It seeks to persuade us to some content or activities and dissuade the us from others. See Ellen Lupton's text on [typographic hierarchy](#).



Grids



Proportion, geometry and grids are rooted from the tradition of early book design (ie. Gutenberg Bible).

In the online space, grids are used to create consistent margins and columns. These grid systems create an underlying structure unifying a page layout. The guidelines in a grid system help the designer align elements in relation to each other and populate content systematically on a web page.

Grid Systems

 <p>The typographic grid is a proportional machine for composition, value, picture, etc. It is a formal programme to accommodate a subjective form. The typographic grid is a proportional machine for composition, value, picture, etc. It is a formal programme to accommodate a subjective form.</p>	<p>Grid systems</p> <p>A grid can be simple or complex, specific or generic, tightly defined or loosely interpreted. Typographic grids are all about control. They establish a system for arranging content within the space of page, screen, or built environment. Designed in response to the internal pressure of content (text, image, data) and the outer edge or frame (page, screen, window), an effective grid is not a rigid formula but a flexible and resilient structure, a skeleton that moves in concert with the muscular mass of content. Grids belong to the technological framework of typography, from the concrete modularity of intertypes to the ubiquitous rulers, guides, and coordinate systems of graphic applications. Although software generates illusions of smooth curves and continuous tones, every digital image or mark is constructed—ultimately—from a grid of neatly bounded blocks. The ubiquitous language of the grid (graphical user interface) creates a guided space in which windows convey windows. In addition to their place in the background of design production, grids have become explicit theoretical tools. Avant-garde designers in the 1950s and 1960s imposed the mechanical grid of intertypes, bringing it to the polemical surface of the page. In Switzerland after World War II, graphic designers built a total</p>	<p>A grid can be simple or complex, specific or generic, tightly defined or loosely interpreted. Typographic grids are all about control. They establish a system for arranging content within the space of page, screen, or built environment. Designed in response to the internal pressure of content (text, image, data) and the outer edge or frame (page, screen, window), an effective grid is not a rigid formula but a flexible and resilient structure, a skeleton that moves in concert with the muscular mass of content. Grids belong to the technological framework of typography, from the concrete modularity of intertypes to the ubiquitous rulers, guides, and coordinate systems of graphic applications. Although software generates illusions of smooth curves and continuous tones, every digital image or mark is constructed—ultimately—from a grid of neatly bounded blocks. The ubiquitous language of the grid (graphical user interface) creates a guided space in which windows convey windows. In addition to their place in the background of design production, grids have become explicit theoretical tools. Avant-garde designers in the 1950s and 1960s imposed the mechanical grid of intertypes, bringing it to the polemical surface of the page. In Switzerland after World War II, graphic designers built a total</p>	<p>Grid systems</p> <p>A grid can be simple or complex, specific or generic, tightly defined or loosely interpreted. Typographic grids are all about control. They establish a system for arranging content within the space of page, screen, or built environment. Designed in response to the internal pressure of content (text, image, data) and the outer edge or frame (page, screen, window), an effective grid is not a rigid formula but a flexible and resilient structure, a skeleton that moves in concert with the muscular mass of content. Grids belong to the technological framework of typography, from the concrete modularity of intertypes to the ubiquitous rulers, guides, and coordinate systems of graphic applications. Although software generates illusions of smooth curves and continuous tones, every digital image or mark is constructed—ultimately—from a grid of neatly bounded blocks. The ubiquitous language of the</p>	<p>A grid can be simple or complex, specific or generic, tightly defined or loosely interpreted. Typographic grids are all about control. They establish a system for arranging content within the space of page, screen, or built environment. Designed in response to the internal pressure of content (text, image, data) and the outer edge or frame (page, screen, window), an effective grid is not a rigid formula but a flexible and resilient structure, a skeleton that moves in concert with the muscular mass of content. Grids belong to the technological framework of typography, from the concrete modularity of intertypes to the ubiquitous rulers, guides, and coordinate systems of graphic applications. Although software generates illusions of smooth curves and continuous tones, every digital image or mark is constructed—ultimately—from a grid of neatly bounded blocks. The ubiquitous language of the</p>	<p>A grid can be simple or complex, specific or generic, tightly defined or loosely interpreted. Typographic grids are all about control. They establish a system for arranging content within the space of page, screen, or built environment. Designed in response to the internal pressure of content (text, image, data) and the outer edge or frame (page, screen, window), an effective grid is not a rigid formula but a flexible and resilient structure, a skeleton that moves in concert with the muscular mass of content. Grids belong to the technological framework of typography, from the concrete modularity of intertypes to the ubiquitous rulers, guides, and coordinate systems of graphic applications. Although software generates illusions of smooth curves and continuous tones, every digital image or mark is constructed—ultimately—from a grid of neatly bounded blocks. The ubiquitous language of the</p>  <p>The typographic grid is a proportional machine for composition, value, picture, etc. It is a formal programme to accommodate a subjective form. The typographic grid is a proportional machine for composition, value, picture, etc. It is a formal programme to accommodate a subjective form.</p>
--	--	---	--	---	--

The grid provides a system for structuring and dividing content.

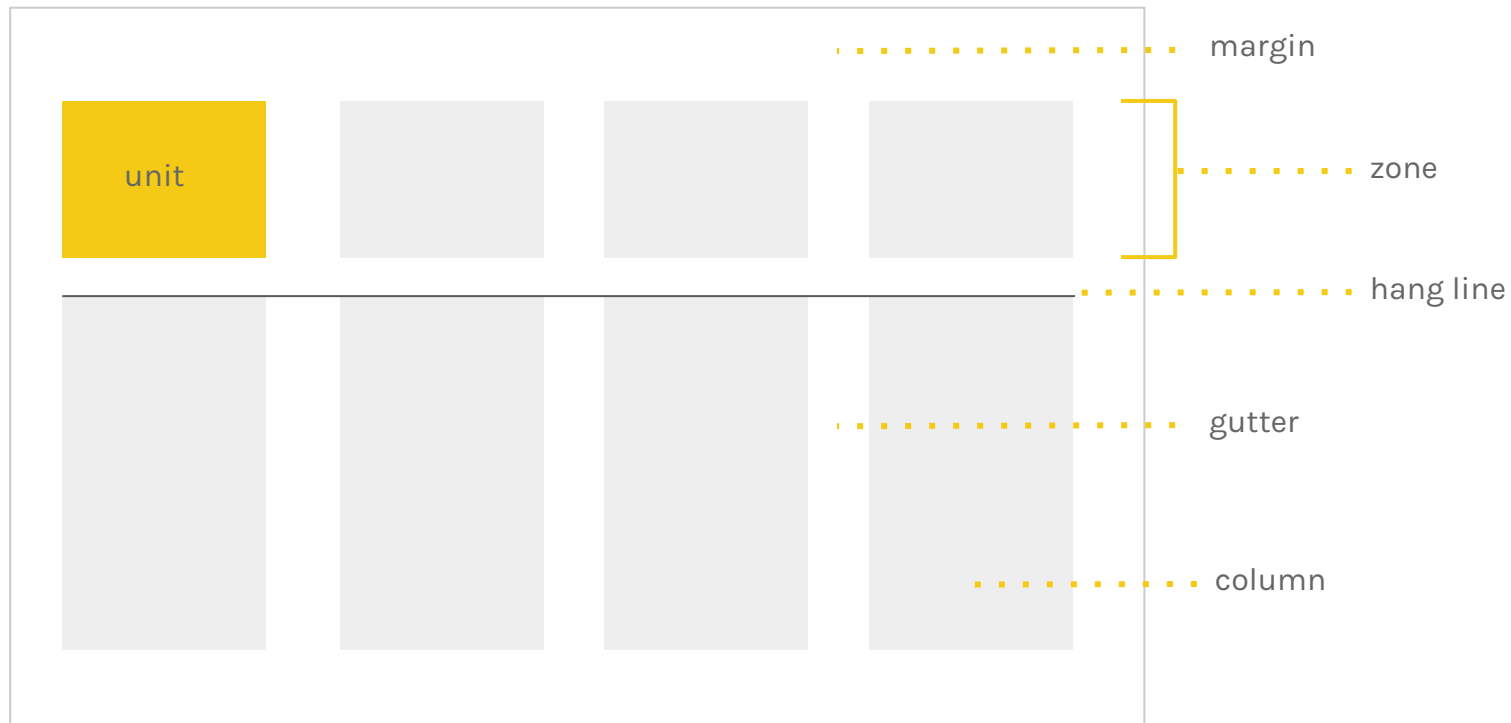
Some online examples:

- [The Grid System](#)
- [Writings by José Manuel Riveros](#)
- [New York Times](#)

Approaches to building grids:

- [Don't Overthink It Grids](#)
- [Content Layout](#)

Components of a Grid



Responsive Design

Keeping your layouts fluid

Responsive

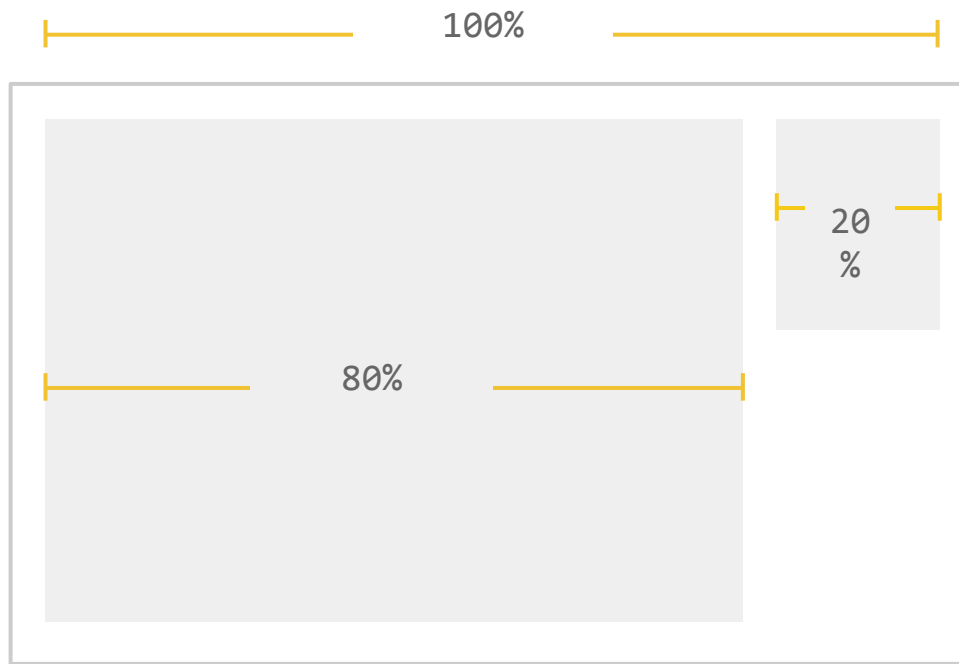
Responsive web design refers to the ability of websites to adapt the presentation of content to a range of devices and media.

The term was coined by [Ethan Marcotte](#).

Responsive design involves:

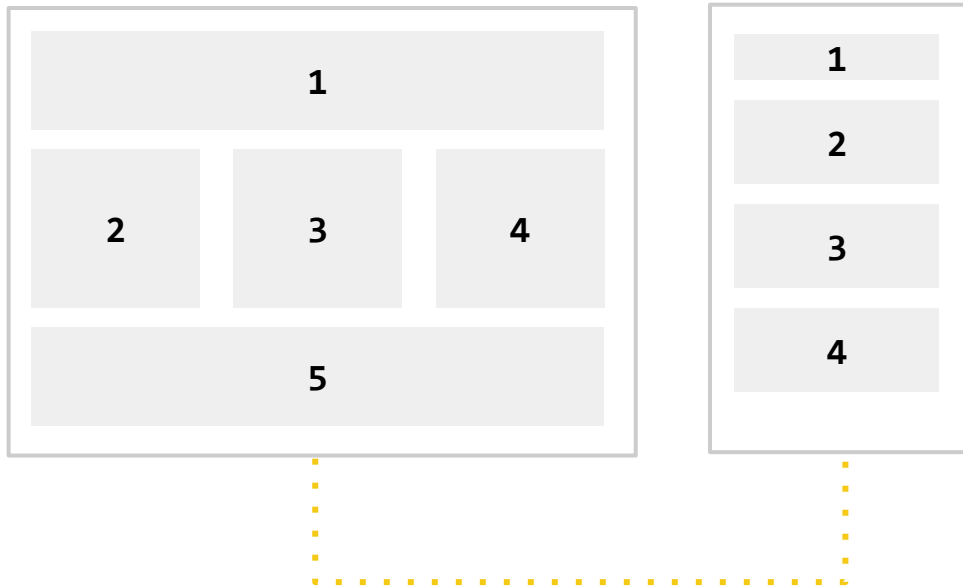
- Fluid Layout
- Fluid Images
- Media Queries

Fluid Layout



- Use percentage widths relative to document body size
- Respond to adjustments in browser widths

Rearrangement / Omission



- Block-level containers are rearranged according to the DOM (HTML structure)
- **Breakpoints** determine the browser widths at which your containers will push down (rather than shrink in width.)
- Some sections may be omitted on the mobile website

Fluid Images

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

```
.container {  
  width: 33%;  
}
```

```
.figure img {  
  width: 100%;  
}
```

Image container has defined, relative width.

The `max-width: 100%;` makes image scale down if it has to, but never scale up to be larger than its original size.

A note on Image formats

Raster / Bitmap

A raster graphics image, or bitmap, is a dot matrix data structure representing a generally rectangular grid of pixels, or points of color, viewable via a monitor, paper, or other display medium. Raster graphics are resolution dependent. They cannot scale up without loss of quality.

Vector

Use geometrical primitives like points, lines, curves and shapes that can be mathematically expressed to represent them in a graphic. Thus vector graphics are not resolution dependent. They can scale up or down without loss in quality.

Image Optimization

Compression: Image compression algorithms seek to eliminate or reduce redundancy in the image data in order to be able to store or transmit data efficiently. Compression essentially comes in two flavors: Lossy or lossless.

In order keep file sizes to a minimum but retain the quality needed, make sure you are saving your images in the right image format.

- Main file formats for raster images on the web: PNG, GIF, and JPG.
- JPG best for photography and detailed line art with subtle shifts in tonal range.
- PNG is best suited for logos and shapes with a few number of colors. PNGs allow for transparency.
- GIF is best for solid blocks of color. Is similar to PNG and allows for transparency, but has animation capabilities.
- SVG, [fonts](#) are good ways to use vectors to the page instead of images

Retina screens

72dpi (dots / inch, in this case pixels) used to be the standard for screen resolutions. But now with the introduction of mobile devices and retina screens, screen resolutions may be much higher.

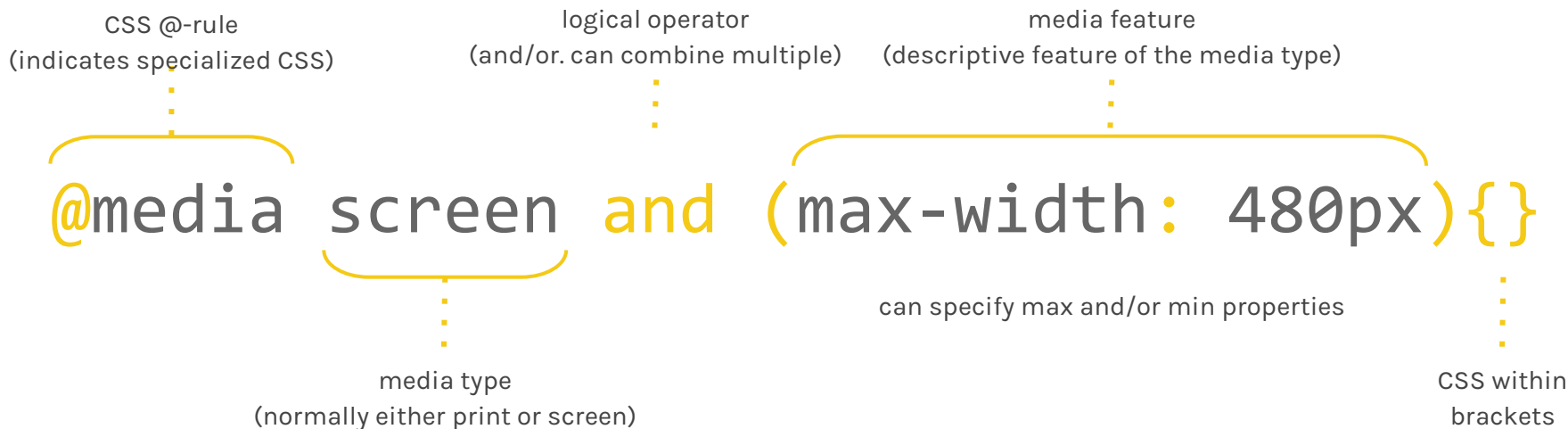
In order to have your images appear crisp on retina screens, you will need to provide higher resolution images (at least 2x large.) Many website prepare multiple versions of the same image so that those on a smaller screen (and a slower connection) will not need to download the higher-res image.

[Read more](#) on various solutions to work with retina images.

Media Queries

Media queries allow designers to tailor CSS for a particular media type and/or media dimensions. These parameters define your **breakpoints**. See [full list](#) of common device breakpoints.

Media queries are usually written at the end of your stylesheet.



Note: Nested Brackets

```
@media screen and (max-width: 480px) {  
    h1 {  
        margin-left: 20px;  
    }  
}
```

Frameworks

Frameworks consist of pre-written code that jumpstarts the development process.

The 960 grid was a popular tool, until responsive layouts became the norm. Today, Bootstrap is widely used.

Introduction to CSS: Review

You should now have an understanding of how to:

- ❑ Use floats
- ❑ Use positioning
- ❑ Create a layout
- ❑ Distinguish between uses of float and positioning
- ❑ Use a grid system
- ❑ Develop a responsive design