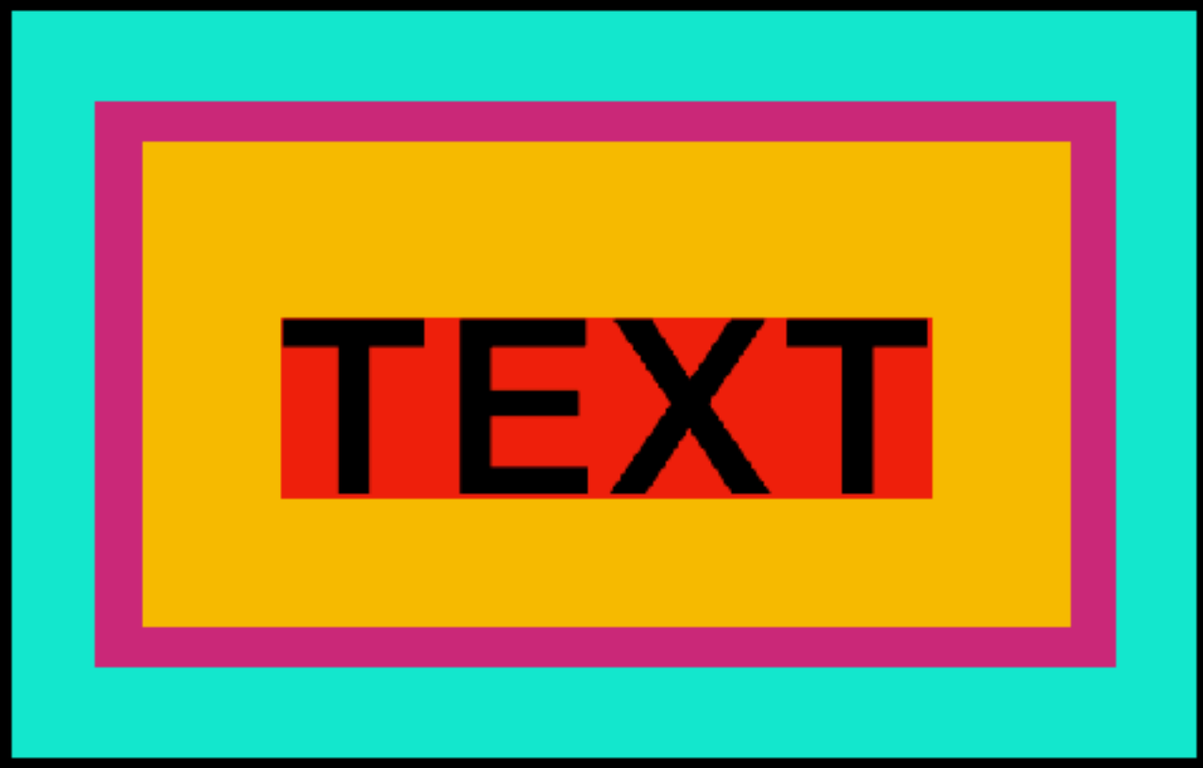
the Box Model

Border

All boxes have borders even if invisible or 0px wide. It separates the edge of one box from another.

Padding

Padding is the space btw the border + any content contained within it. More padding increases the readability of its contents.



Margin

Margins sit outside the edge of the border. You can set the width to create a gap btw borders of adjacent boxes.

Content

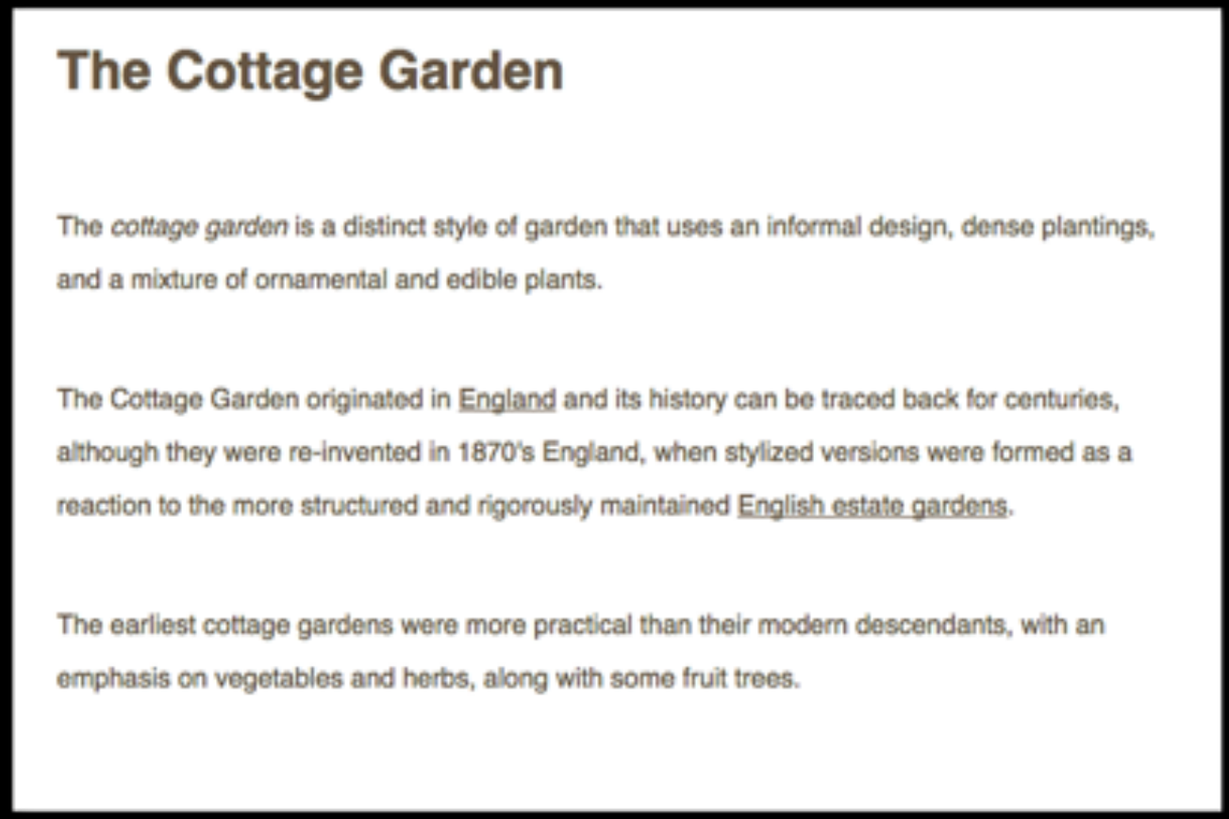
HTML - Hyper Text Mark Up

is a grammar for structuring web pages. It defines paragraphs, headings, data tables + media elements. HTML describes the content of the page - not how it looks.

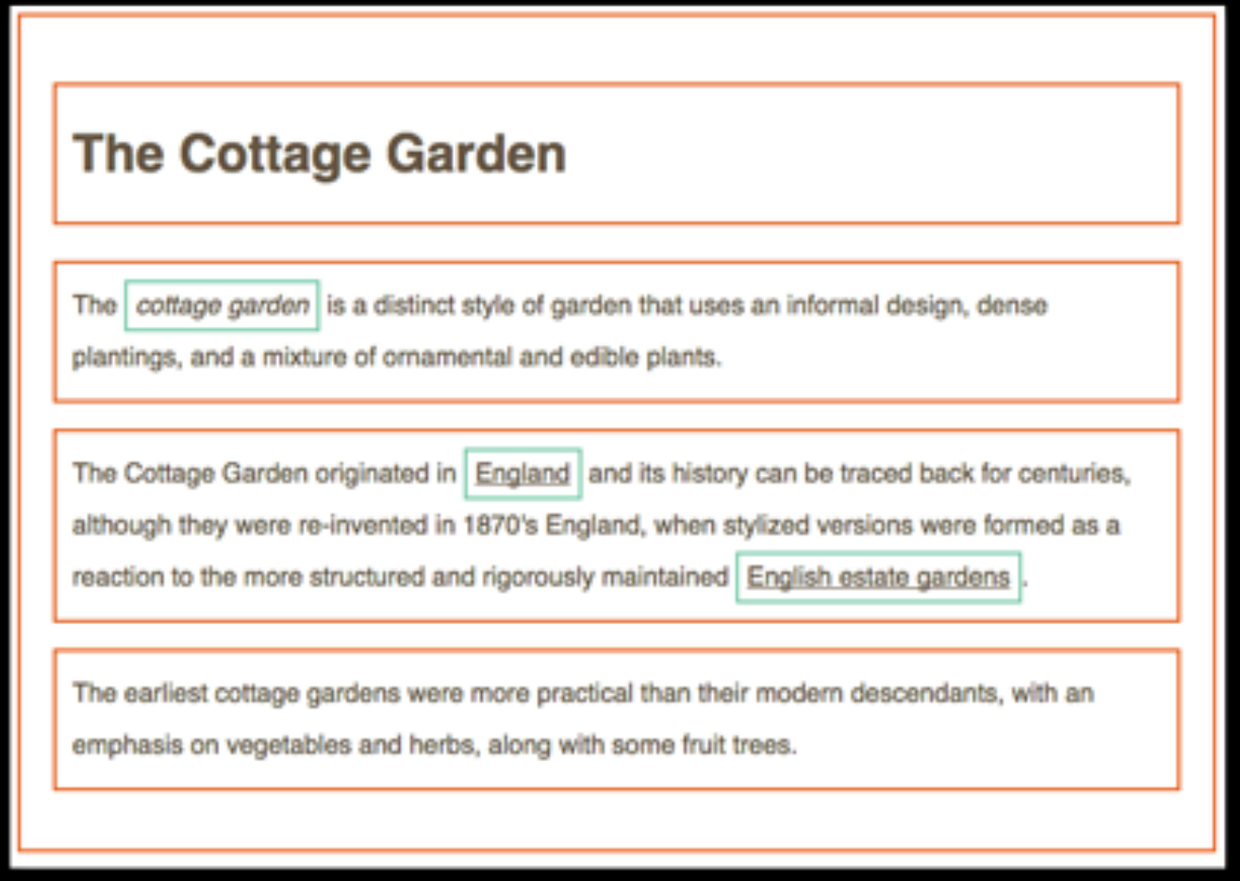
CSS - Cascading Style Sheet

rules for styling a web page. Setting colors, typeface, and the layout. It can be used to consider the design of your **page across different platforms and screen sizes.**

The key to understanding how **CSS** works is to imagine that there is an invisible box around every **HTML** element.



Block level elements are outlined w/ red + inline elements in green.



<body> creates 1st box, then **<h1>**, **<h2>**, **<p>**, **<i>** + **<a>** each create their own boxes within it.

Quick Getting Ahead of Ourselves: responsive web design

Metadata: `viewport`

The user's visible area of a web page

HTML5 introduced a method to let web designers take control over the viewport, through the **<meta>** tag.

```
<!
—  - Tells the browser to match the device's width for the viewport
     - Sets an initial zoom value -->

<meta name="viewport" content="width=device-width, initial-scale=1.0”>
```

<meta name="viewport" content="width=device-width, initial-scale=1.0”>

without

with

Selector	Meaning	Example
Universal Selector	Applies to all elements in the document	* { }
Type Selector	Matches element names	h1, h2, h3 { }
Class Selector	Matches an element whose class attribute has a value that matches the one specified after the period (or full stop) symbol	.theNote { } targets any element whose class attribute has a value of "note}
		p.note { } targets only <p> elements whose class attribute has a value of "note"
ID Selector	Matches an element whose id attribute has a value that matches then specified after the # symbol	#introduction { } targets the element whose id attribute has value of "introduction"

Selector	Meaning	Example
Child Selector	**Matches an element that is a direct child of another**	li > a { } targets any <a> element that are children of an element (but not other <a> elements in the page.
Descendant Selector	**Matches an element that is a descendent of another specified element (not just a direct child of that element)**	p a { } targets any <a> elements that sit inside a <p> element, even if there are other elements nested btw them

Selector	Meaning	Example
Adjacent Sibling Selector	**Matches an element that is the next sibling of another**	**h1+p** { } targets the first **<p>** element after any **<h1>** element (but not other **<p>** elements)
General Sibling Selector	**Matches an element that is a sibling of another, although it does not have to be the directly preceding element**	**h1~p** { } tif you have two **<p>** elements that are siblings of an **<h1>** element, this rule would apply to both

```css
/* type/element selector */
p {
    color: blue;
    font-size: 50vh;
}

/* class attribute selector */
.myBlueText {
    color: blue;
}

/* id attribute selector */
#blue-par {
    color: blue;
}

/* BONUS: grouping
selector */
p,
.blue-text,
#blue-par {
    color: blue;
}
```

selecting multiple elements:

```
h1, h2, h3 {

    color: red;
    background-color: blue;
    width: 500px;

}
```

```
p,
li {

    background-color: red;
    font-color: blue;


}
```

HTML comments are written like this

<!-- This is a comment -->

CSS comments are written like this

/* This is a comment */

```
 {
text-align:

            left ;
            right ;
            center ;
            justify ;

}
```

```
{
vertical-align:

        baseline ;
        sub ;
        super ;
        top ;
        text-top ;
        middle ;
        bottom ;
        text-bottom ;


}
```

This property is NOT intended to allow you to vertically align text in the middle of a block level elements such as <p> + , although it does have this effect when used with table cells <td> + <th> elements.

It is more commonly used w/ inline elements such as <img>, or . When used with these elements, it performs a task very similar to the HTML align attribute used on the <img> element.

Interaction Design

```
a: link {

a: visited {

    : hover {

    : active {

    : focus {

}
```

Applied when a user hovers over an element w/ a mouse. This changes the appearance of links and buttons when a user places their cursor over them. Does not work on mobile.

Applied when an element is bingo activated by a user, like when a button is pressed or a link clicked. This added to UX.

Applied when an element has focus. Any thing you can interact with.

Focus occurs when a browser discovers that you are ready to interact w/ an element. For example when yr cursor is in an input - that element is said mohave focus.

Classes and IDs

Two common attributes used to single out certain HTML elements are **class** and **id**, both are used to identify particular elements when adding CSS styling rules.**You author class + id names!!** They have no particular meaning in themselves, besides a puzzle - or code - you are creating.

Use a **class** when you have more than one element you want to share the same styling - perhaps across multiple pages.

Use an **id** when there is only one element on the page with that id, for example id=”header”
With a class you can have as many elements with that styling as you like.

An element can have more than one **class**, but not more than one **id**. When there is more than one class, the class names are separated by spaces.

```
<h1 id="myHeader">Hello World!</h1>
```

IDs

Every HTML element can carry the id attribute. It is used to uniquely identify that element from other elements on the page.

Its value should start with a letter or an underscore (not a number or any other character). It is important that no two elements on the same page have the same value for their id attributes (otherwise the value is no longer unique).

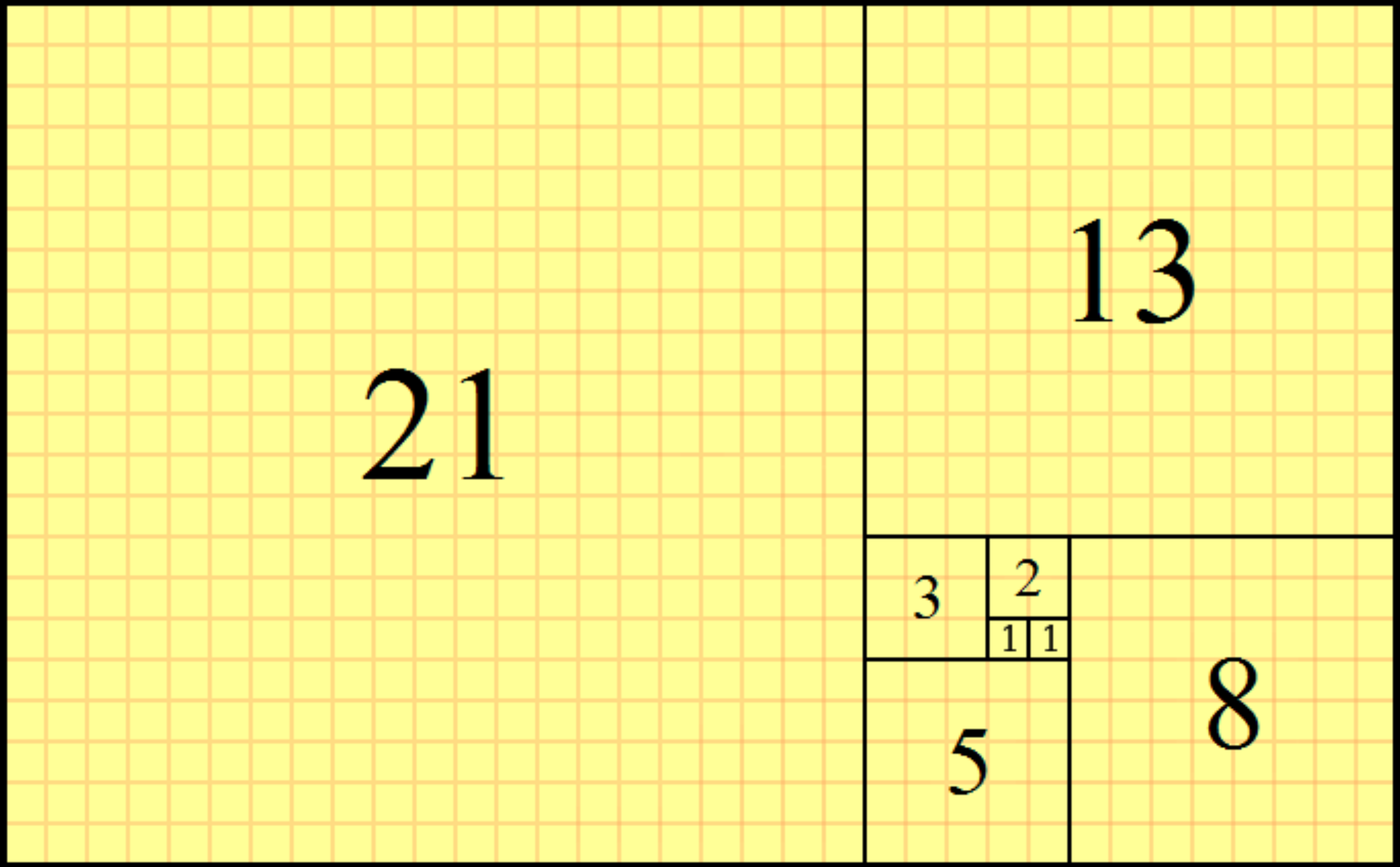
More to read on ID naming: https://mathiasbynens.be/notes/css-escapes

IDs

To select these IDs in CSS you would do so with **#myHeader** syntax

(IDs may become particularly useful when it comes to media elements - photos, videos + sound files.)

```
#myHeader{
    color: blue;
}
```



Fibonacci Sequence
1:1.61 ratio, the Golden Ratio

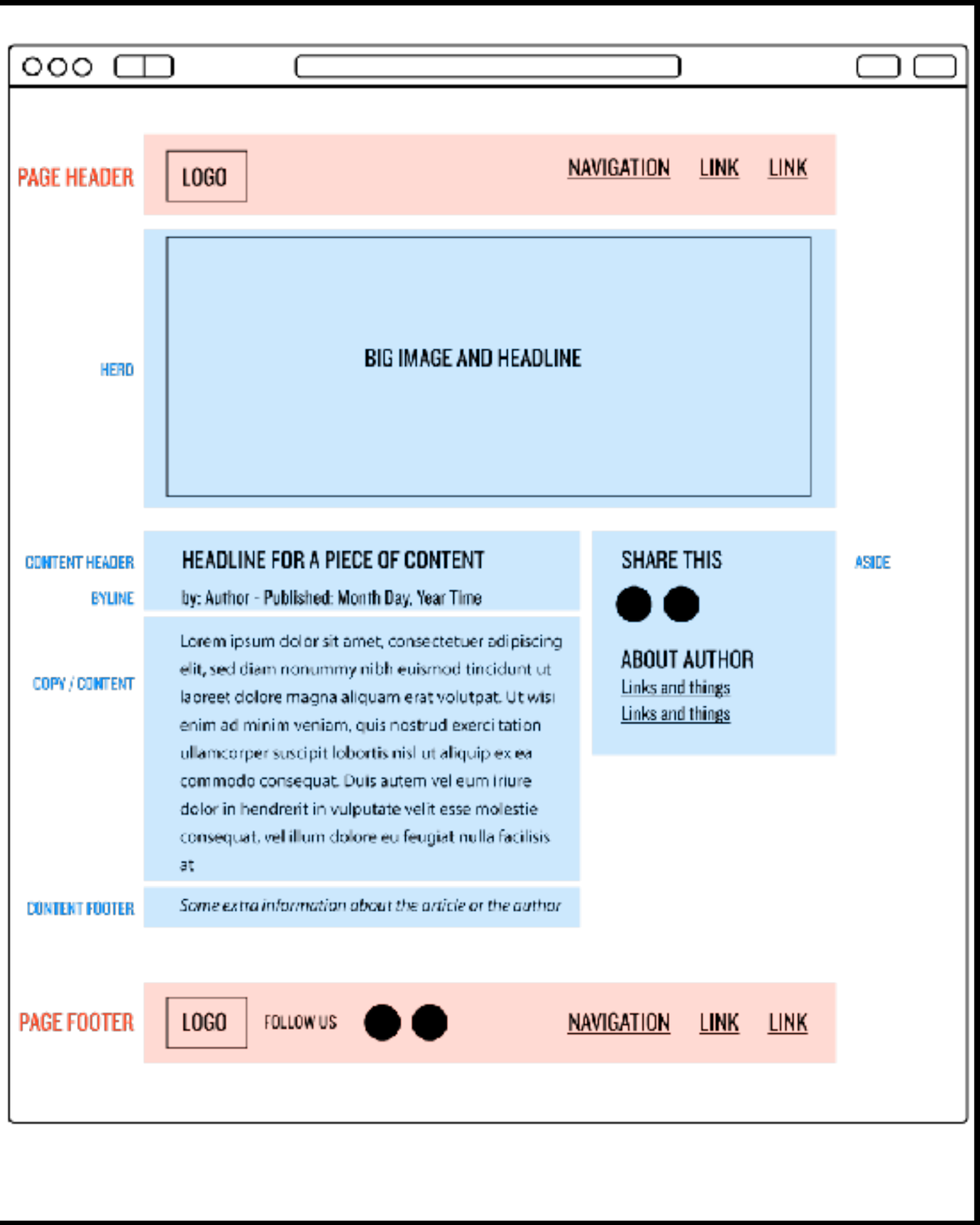
Display Property

```
display: none;   —- html elements default visible

                 —- override default html position
display: inline;
display: block;

          —- responsive way to deal with positioning
display: flex;
display: grid;
```

css layout



— Learn Enough CSS + Layout

So basically up until now I've instructed to do things a particular way. Bc w/ html, git, unix, etc there is only one way to do something (or a piece of software over the process). W/ CSS - there is no "right" answer. When designing websites many solutions to yr problem will exist - which means subjective judgment is the rule rather than the exception.

— Learn Enough CSS + Layout

You have to get used to the idea that no site is going to be exactly the same when viewed by different people. You'll learn to design (or implement other people's designs) in a way that allows room for CSS's inherent ambiguity. Unlike the tightly constrained world of print design, getting things to look exactly the same in every browser and on every operating system is just something you have to give up worrying about.

— Learn Enough CSS + Layout

Font stack

It’s important to understand that the browser will only display font if it’s installed on user’s computer.

Font stack - a collection of more than one typeface in an order of preference to be displayed in the browser if some of the typefaces are not found.

```
{
  font-family: Georgia, Courier, serif;
}
```

Font

Padding is the space btw the border + the content.

Some Properties:

- font-family
- color
- font-size
- line-height
- text-align

Font

Padding is the space btw the border + the content.

text-decoration — underline, strike thru or none (eg to unset underline on hyperlinks

text-transform — change font **case** (eg uppercase, lower, capitalize, none)

font-style — set to italic or normal

font-weight — set to bold or normal

letter-spacing — controls the space btw letters

font-family property sets the font in your CSS

Presented as a hierarchy of choices (1st choice, 2nd choice, 3rd choice) so it's good to have a fallback for older browsers that can't render

```
body {
    font-family: Georgia, Courier, serif;
}
h1, h2, h3 {
    font-family: Arial, Verdana, sans-serif;
}
```

Custom web fonts: Google Fonts

Add link in **<head>** of HTML

```
<link href="https://fonts.googleapis.com/css?
family=Roboto" rel=“stylesheet">
```

Use with font-family property in CSS

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