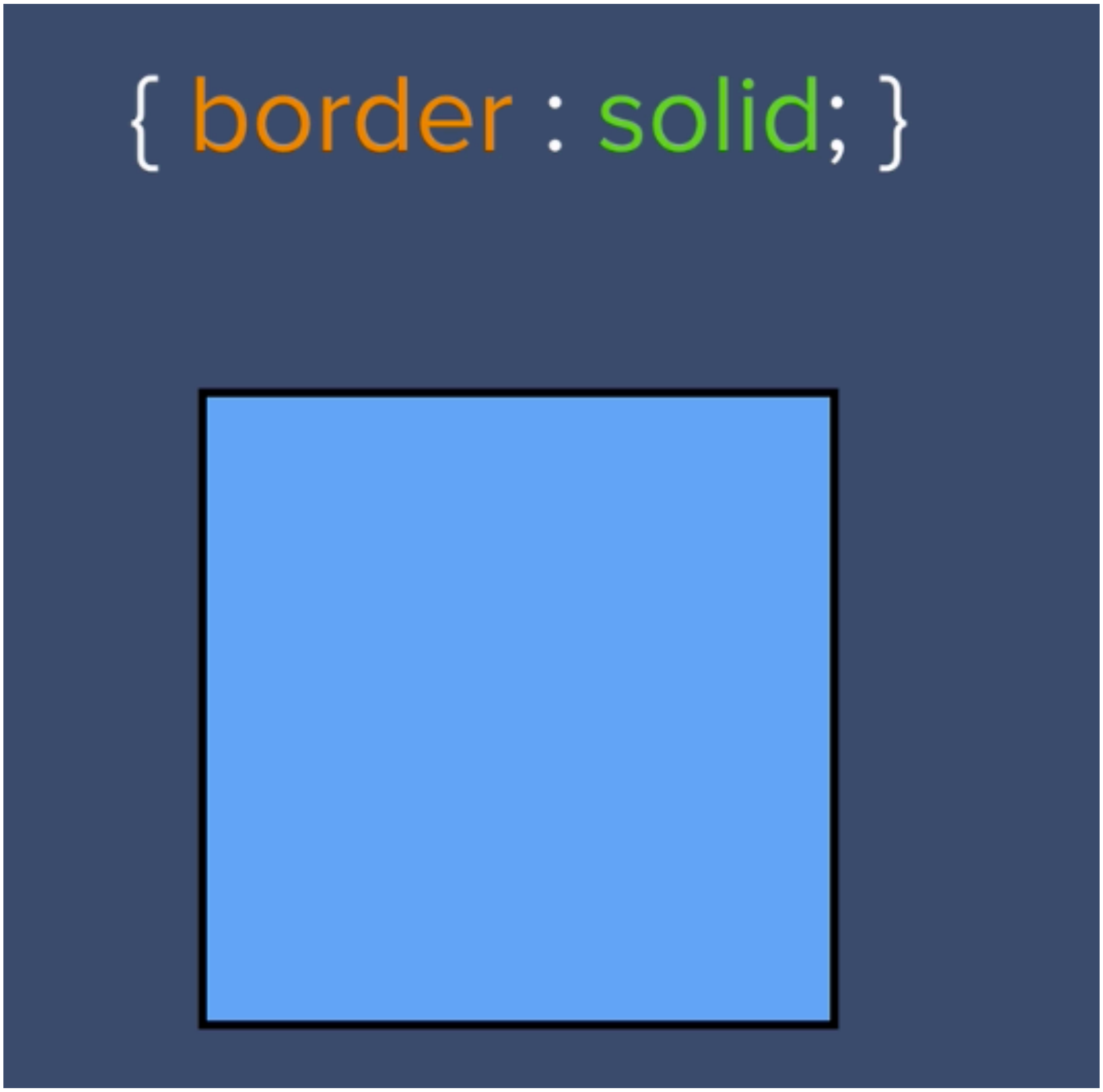
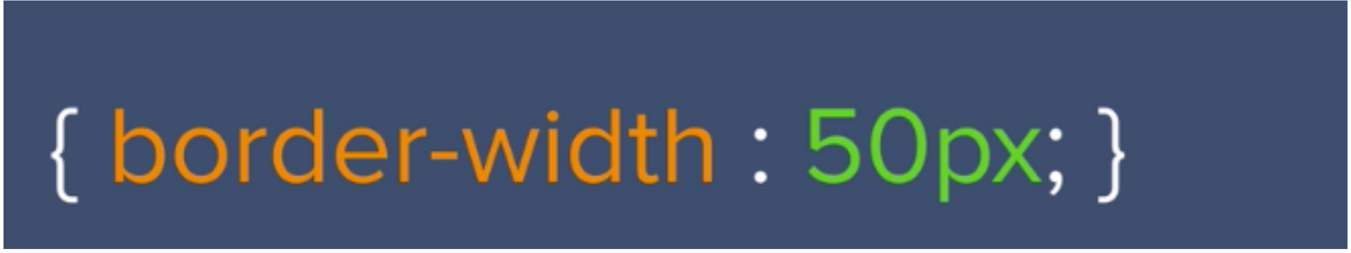
Favicon.cc: generate your favicon
Div: divide up so you can structure the website, content division element.
Group together sth, so you can design it different than the others
Margin Before is at top
Margin after is at bottom
Margin start is at left
Margin end is at right



```
{ border : solid; }
```

The pixel doesn’t change. But you can change the pixel by specifying, make it 50px. So the entire is 400px

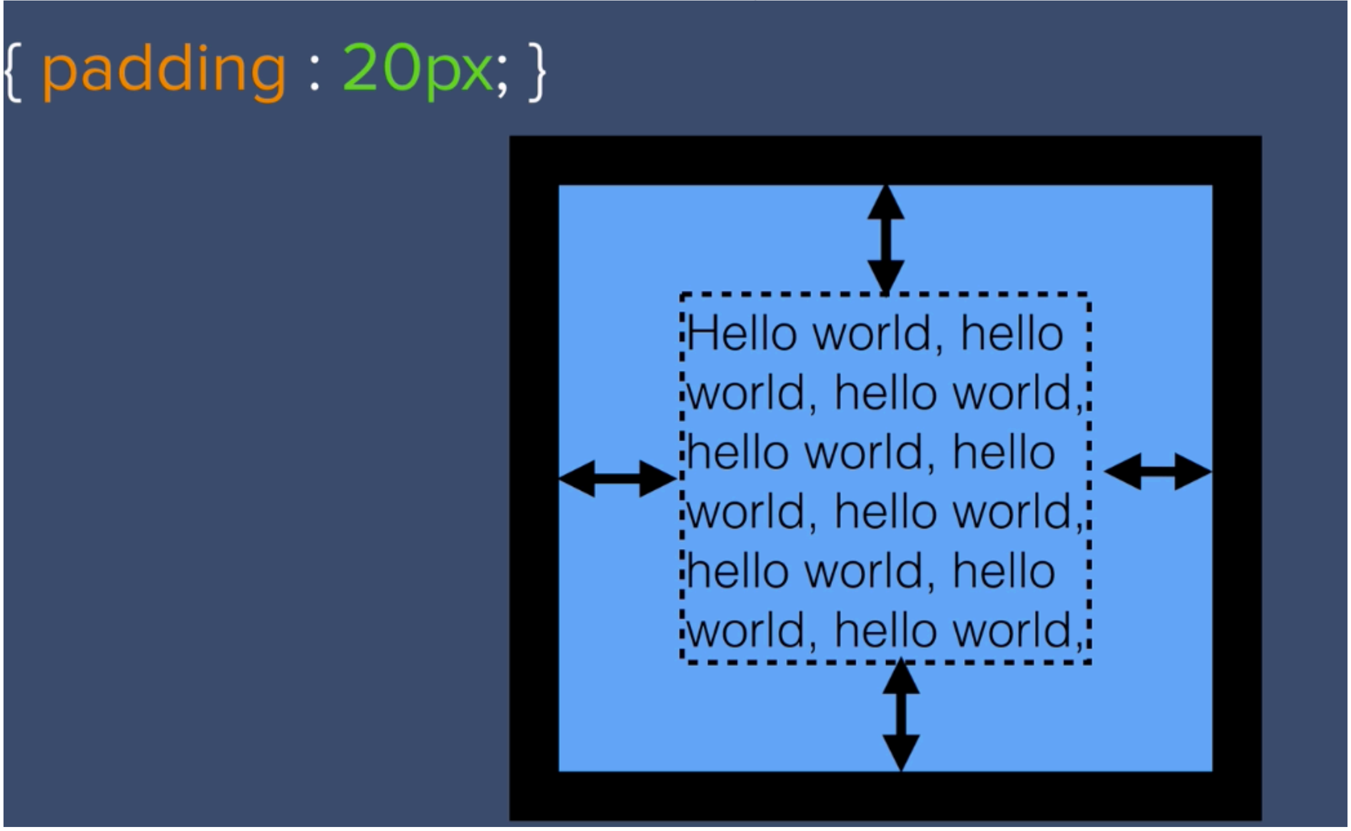


```
{ border-width : 50px; }
```

You could also specify: 0 at top, 10 at right, 20 at bottom, 30 at the left (clockwise specification)

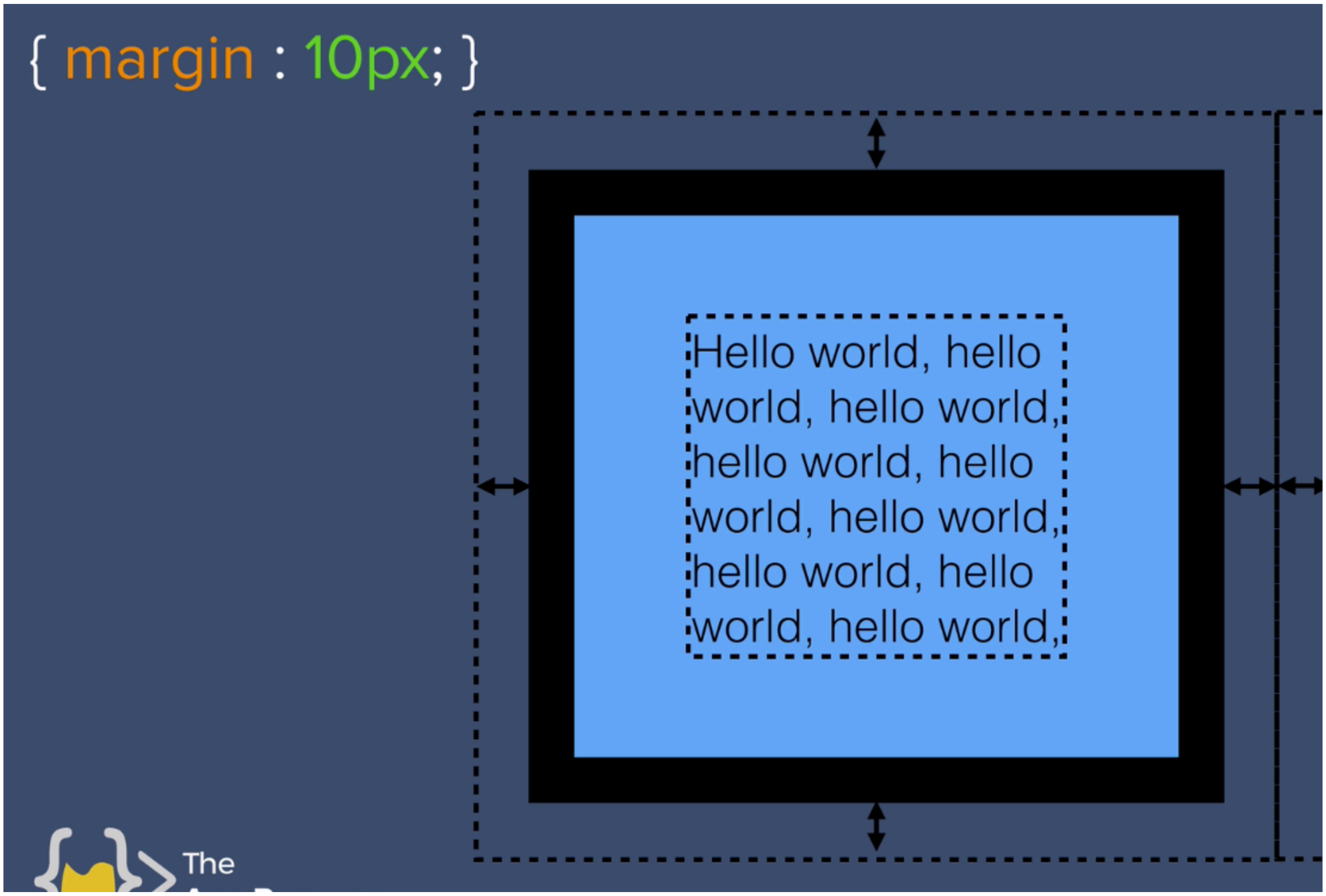
```
{ border-width : 0px 10px 20px 30px; }
```

If you have text in the box, you can specify and have paddings:

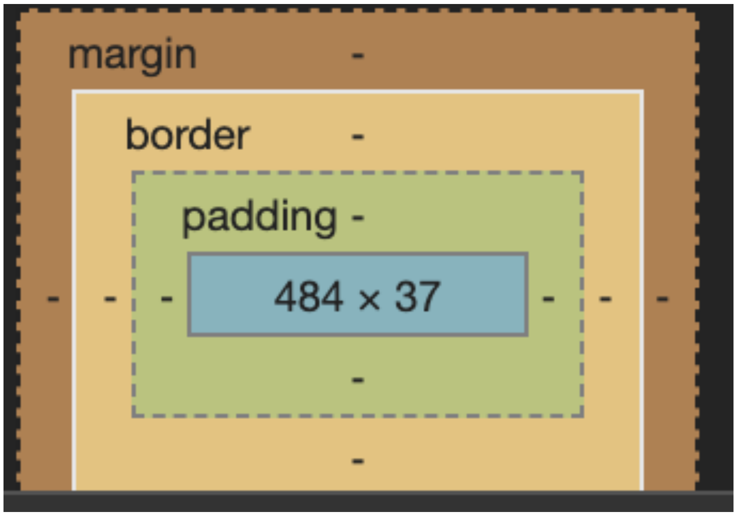


```
{ padding : 20px; }
```

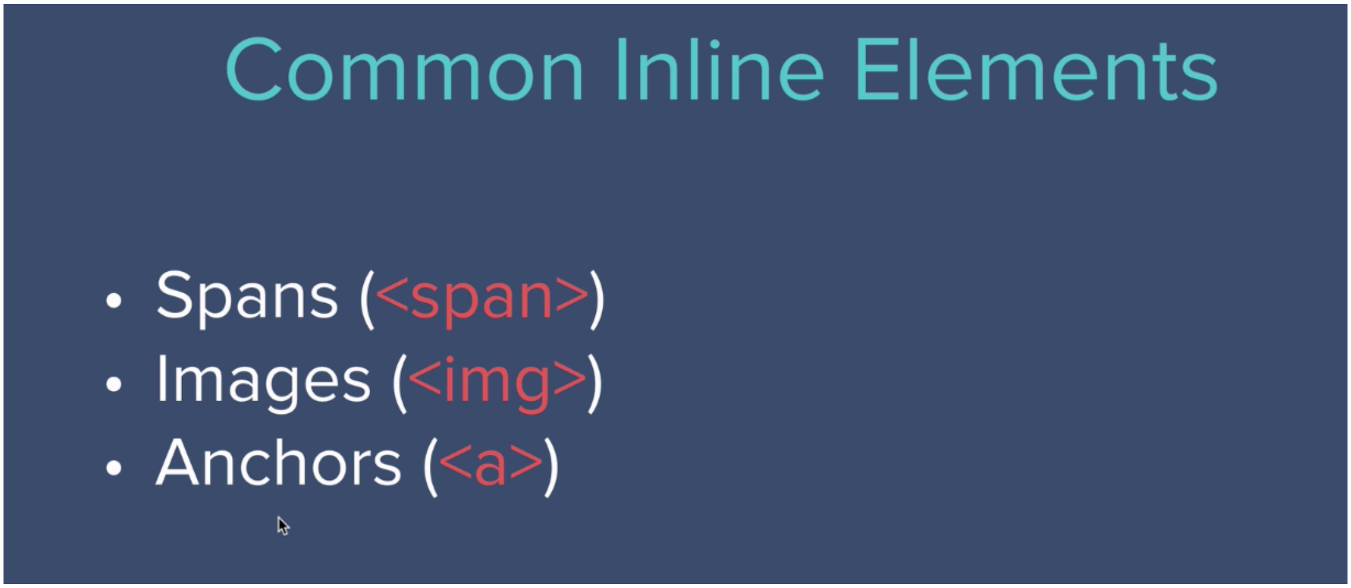
The box is now larger. You can also add margin, light a buffer zone.


{ margin : 10px; }
Hello world, hello world, hello world, hello world, hello world, hello world, hello world, hello world, hello world,
The

So it goes: Margin -> boarder -> padding -> content



Inline elements:



Common Inline Elements

- Spans (<span>)
- Images (<img>)
- Anchors (<a>)

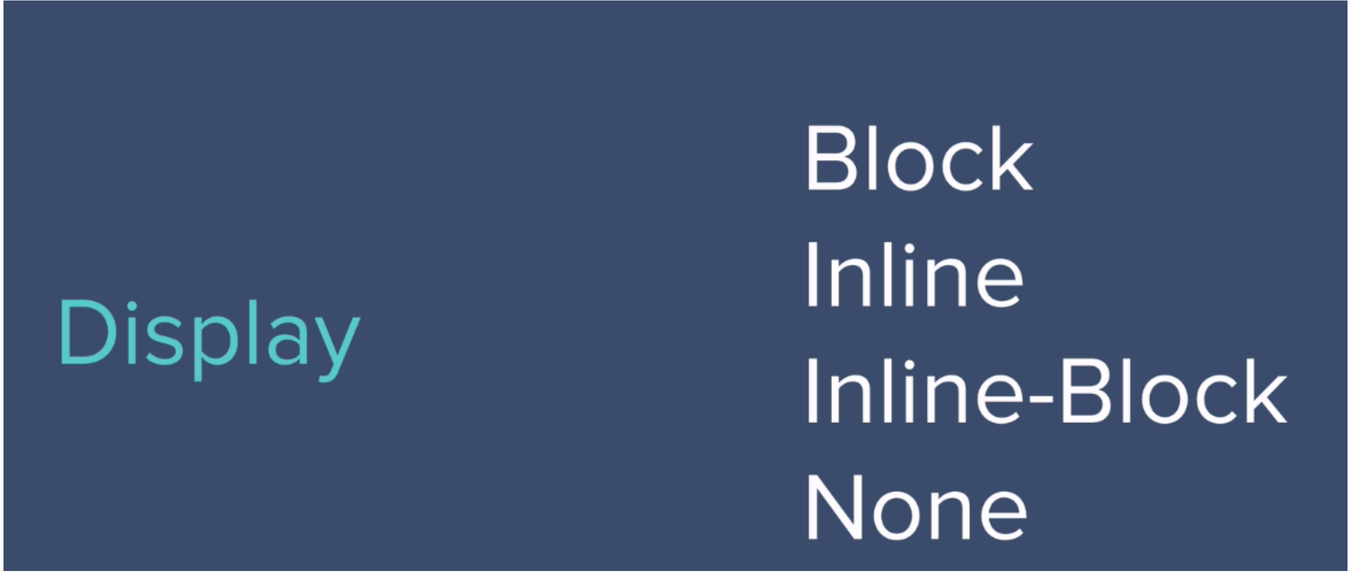
With in line element, you can’t change the width.
But you can specify the width of block element, but other elements can’t seat on the same line.
But you can change this: for example:

```
P{
Display:inline;
}
```

This allow everything to seat next to each other.
So if you want sth that could seat inline but also change their width? You can use In-line Block display:

```
P{
Display:inline-block;
}
```

So for dispaly: None: just get rid of the element.
Other is visability: hidden, it just can’t be seen, but you will still have the empty space.

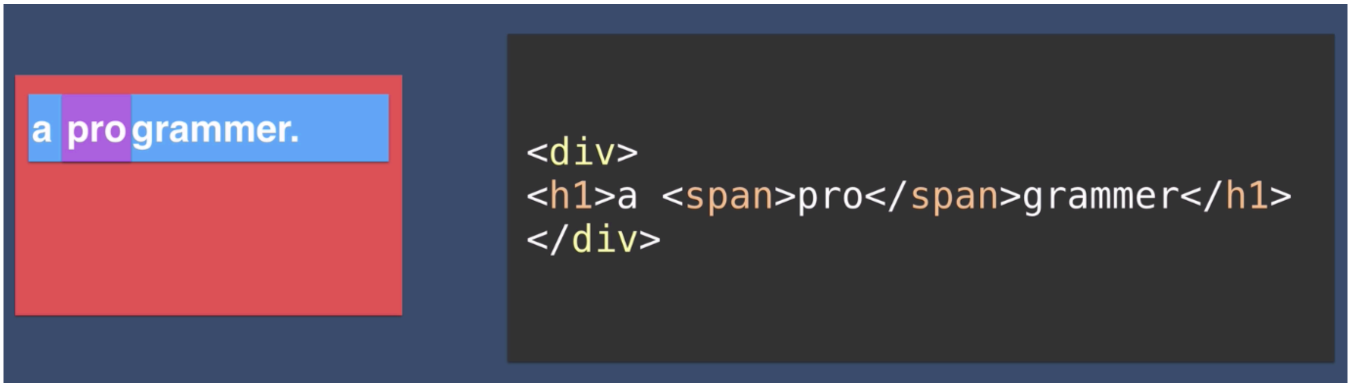


Display

Block
Inline
Inline-Block
None

CSS positioning:
Your HTML already have rules of how to display.

Children sit on top of Parent:



```
<div>
<h1>a <span>pro</span>grammer</h1>
</div>
```

H1 seats on top of the div.
span is ontop of h1

Position with CSS property:
-Static (default html flow)
-Relative (position: relative ◦ left:30px), => (give that image a 30px space, there is top bottom, left and right. Important thing is that when you move a element : it doesn't affect other position). It's relative position to where the original html form
-Absolute (position: absolute ◦ left:30px) => so the position is relative to it's parent. In this case, it would be the parent which is div. Absolute position looks like it's relative to entire page. This however, would affect other elements
-Fixed (position: absolute ◦ left:30px) => No matter how you scroll the webpage, it stays there

Two major fonts:
Sans- serif vs serif
For code: monspace
Cursive (hand written) and fantisy are hard to read.
Websafe fonts:
https://www.w3schools.com/cssref/css_websafe_fonts.asp
https://fonts.google.com/
Images:
https://www.flaticon.com/
Animated image: (use sticker tags, without backgrounds)
https://giphy.com/

To dynamic sites: use em or % for your fonts.
Use REM the least error prone, the size is relative to the root.

CSS float :
Float left in image. The image will be on the left.

CSS clear: think as anti-float

Only use it when you need to wrap text around sth, don't use it to position things ! You will get into many troubles if you abuse it.

CSS bottom generator:
https://cssbuttongenerator.com/