

CSS





WHAT IS CSS ?

CSS Stands for Cascading Style Sheet.,
and It's used to style HTML (Hyper Text
Markup Language).

HTML



CSS



Ways to Add CSS :-

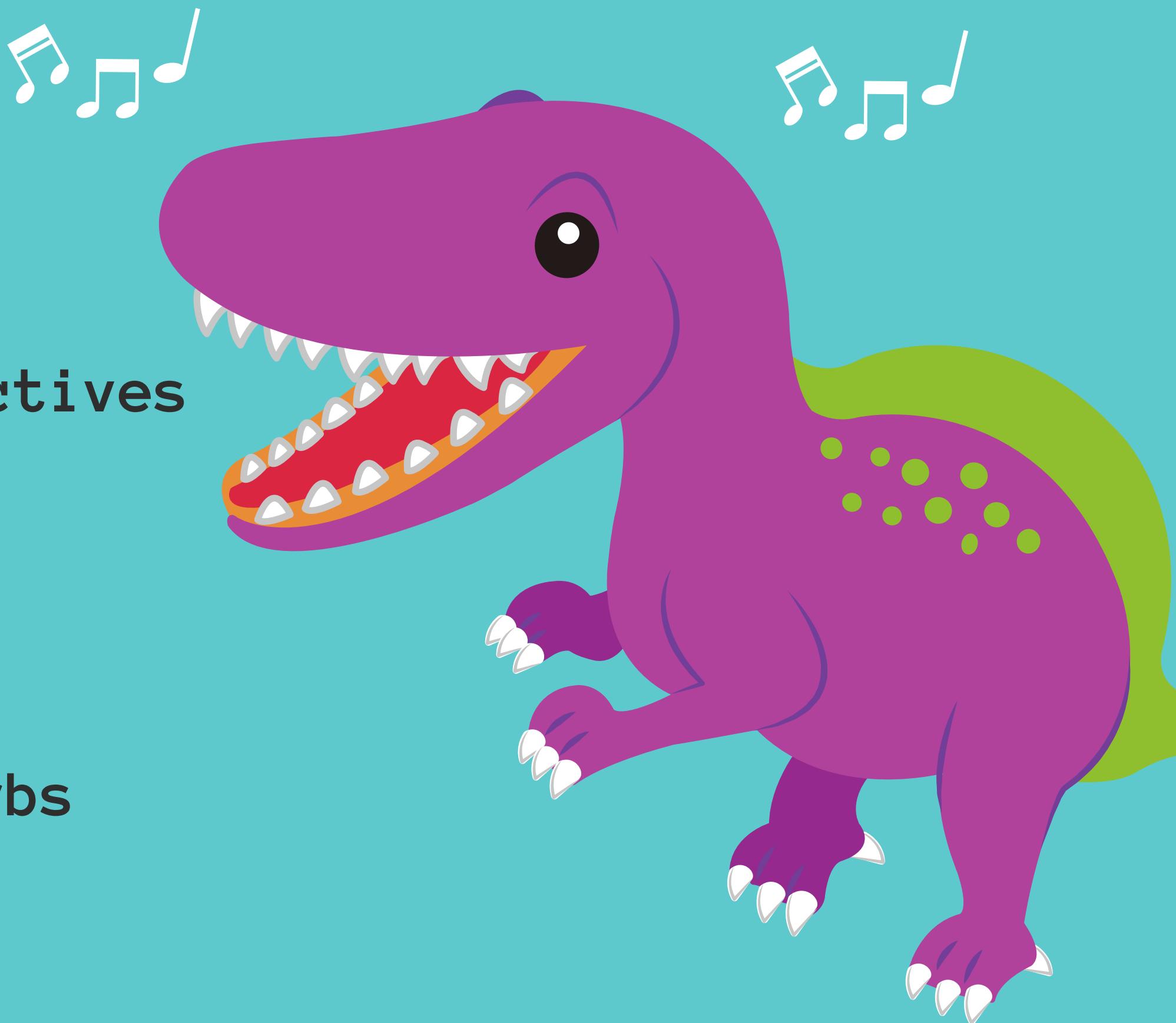
Inline CSS

Internal CSS

External CSS



THE PURPLE CSS – adjectives
DINO HTML – nouns
DANCED JS – verbs





CSS

WHAT IS IT?

CSS is a language for describing how documents are presented visually - how they are arranged and styled.

WHAT DOES IT STAND FOR?

CSS stands for Cascading Style Sheets. We'll cover the "cascading" part in a bit; don't worry about it for now!

THERE'S A LOT!

CSS is very easy to get the hang of, but it can be intimidating because of how many properties we can manipulate.

CSS RULES

(almost) everything you do in CSS follows this basic pattern:

```
selector {  
    property: value;  
}
```

CSS RULES

Make all `<h1>` elements purple

```
h1 {  
    color: purple;  
}
```

CSS RULES

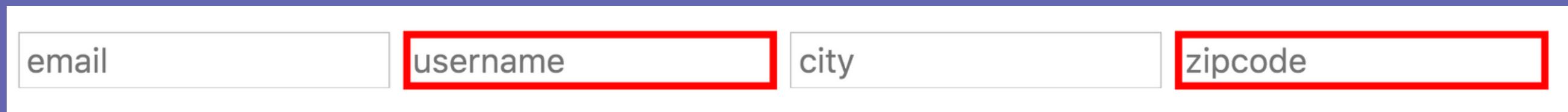
Make all image elements
100 pixels wide & 200 pixels tall

```
img {  
    width: 100px;  
    height: 200px;  
}
```

FANCIER!

Select every other text input
and give it a red border:

```
input[type="text"]:nth-of-type(2n){  
    border:2px solid red;  
}
```

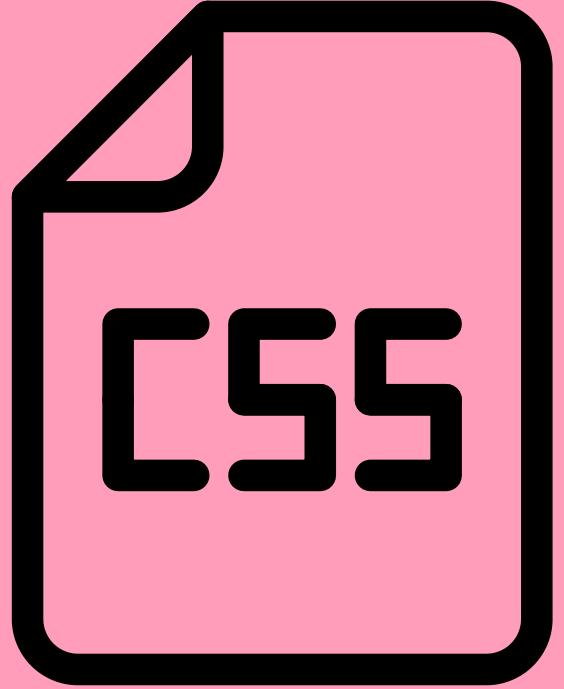


email username city zipcode

So Many CSS Border Properties



border border-blockborder-block-color border-block-end border-block-end-color border-block-end-style border-block-end-width border-block-start border-block-start-color border-block-start-style border-block-start-width border-block-style border-block-width border-bottom border-bottom-color border-bottom-left-radius border-bottom-right-radius border-bottom-style border-bottom width border-collapse border-color border-end-end-radiusborder-end-start-radiusborder-imageborder-image-outsetborder-image-repeatborder-image-sliceborder-image-sourceborder-image-widthborder-inlineborder-inline-colorborder-inline-endborder-inline-end-colorborder-inline-end-styleborder-inline-end-widthborder-inline-startborder-inline-start-colorborder-inline-start-styleborder-inline-start-widthborder-inline-styleborder-inline-widthborder-leftborder-left-colorborder-left-styleborder-left-widthborder-radiusborder-rightborder-right-colorborder-right-styleborder-right-widthborder-spacingborder-start-end-radiusborder-start-start-radiusborder-styleborder-topborder-top-colorborder-top-left-radiusborder-top-right-radiusborder-top-styleborder-top-widthborder-width



Including Styles

INLINE STYLES

You can write your styles directly inline on each element, but this is **NOT A GOOD IDEA** most of the time.

THE <STYLE> ELEMENT

You can write your styles inside of a `<style>` element. This is easy, but it makes it impossible to share styles between documents. **Not recommended either.**

EXTERNAL STYLESHEET

Write your styles in a `.css` file, and then include the using a `<link>` in the head of your html document. **Recommended!**

<link>



```
<head>
  <title>Forms Demo</title>
  <link rel="stylesheet" href="my_styles.css">
</head>
```



css colors



**NAMED
COLORS**

Hotpink

Mediumorchid

Firebrick

Darkkhaki

Gold

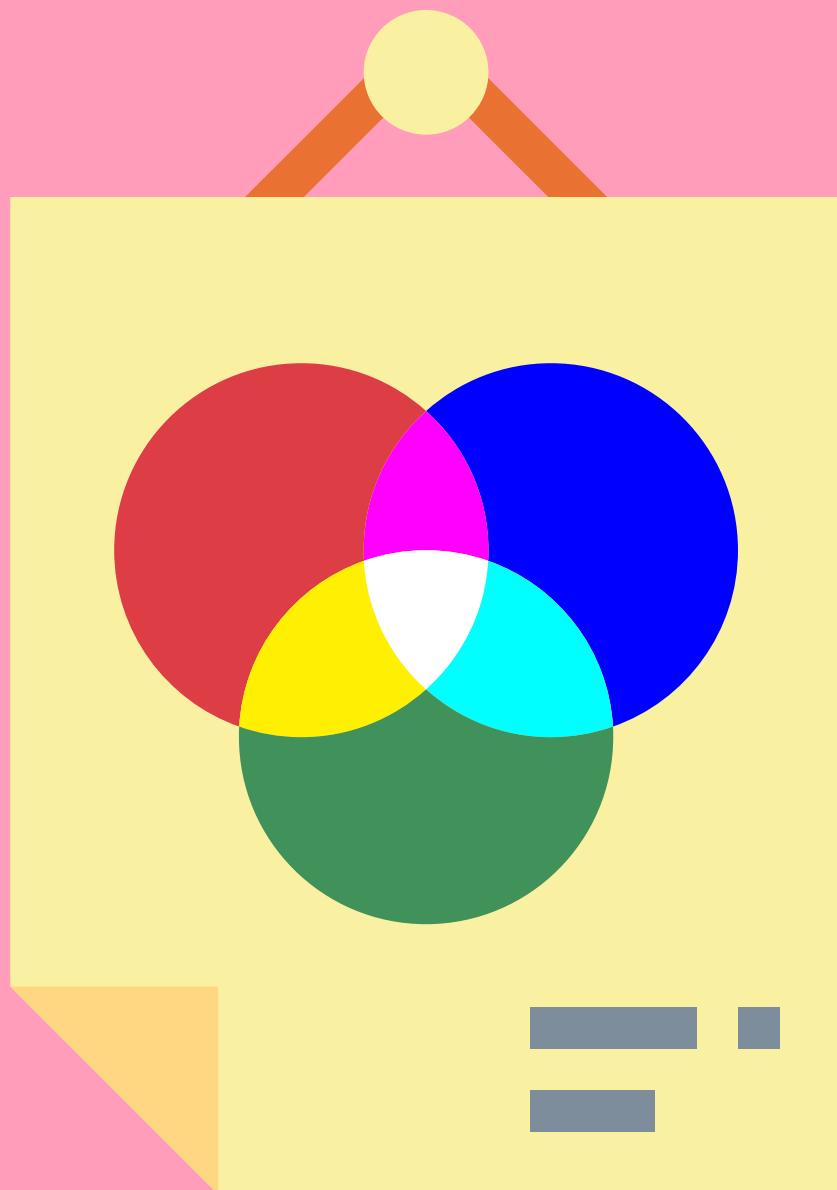
MediumAquamarine

Lightskyblue

Tomato

A typical computer can display
~16,000,000
different colors





RGB

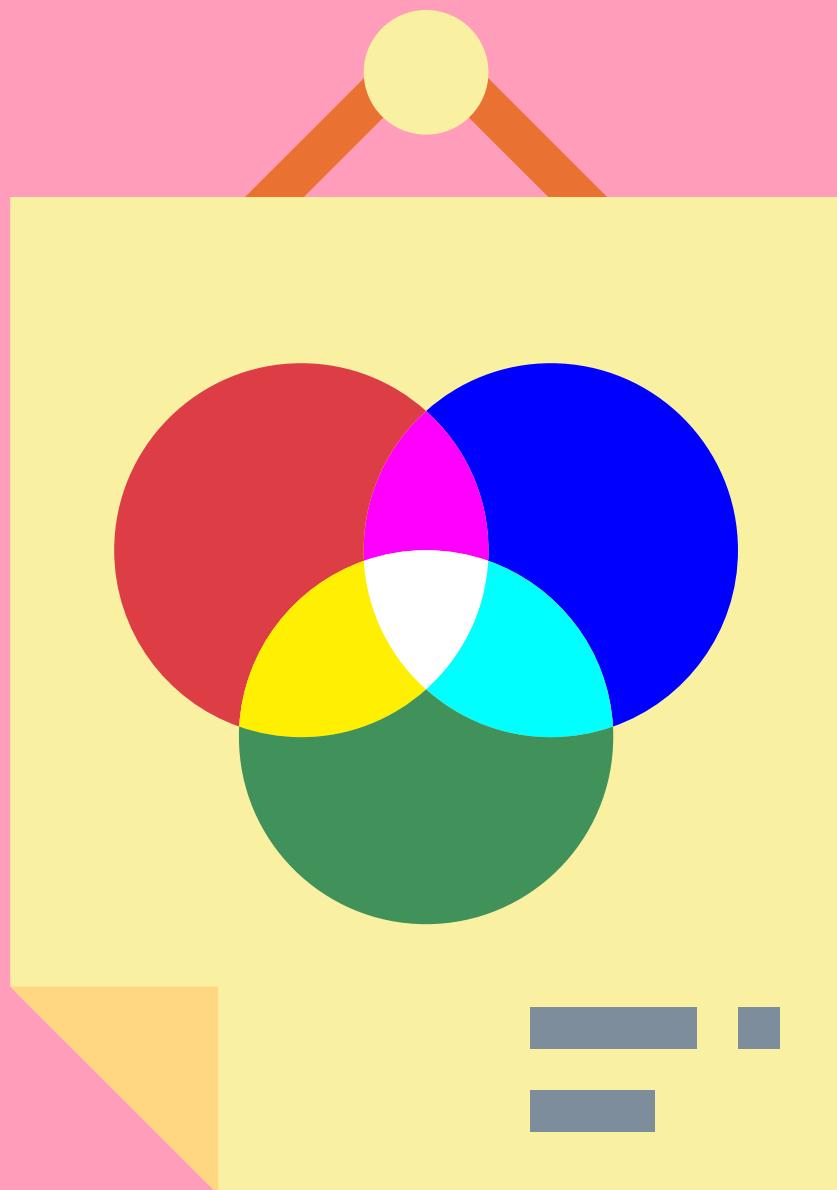
- Red, Green, Blue Channels
- Each channel ranges from 0-255

rgb(255,0,0)

rgb(0,0,255)

rgb(173, 20, 219)

rgb(0,0,0)



Hex

- Still red, green, and blue channels
- Each ranges from 0-255 BUT represented with hexadecimal

Decimal

0,1,2,3,4,

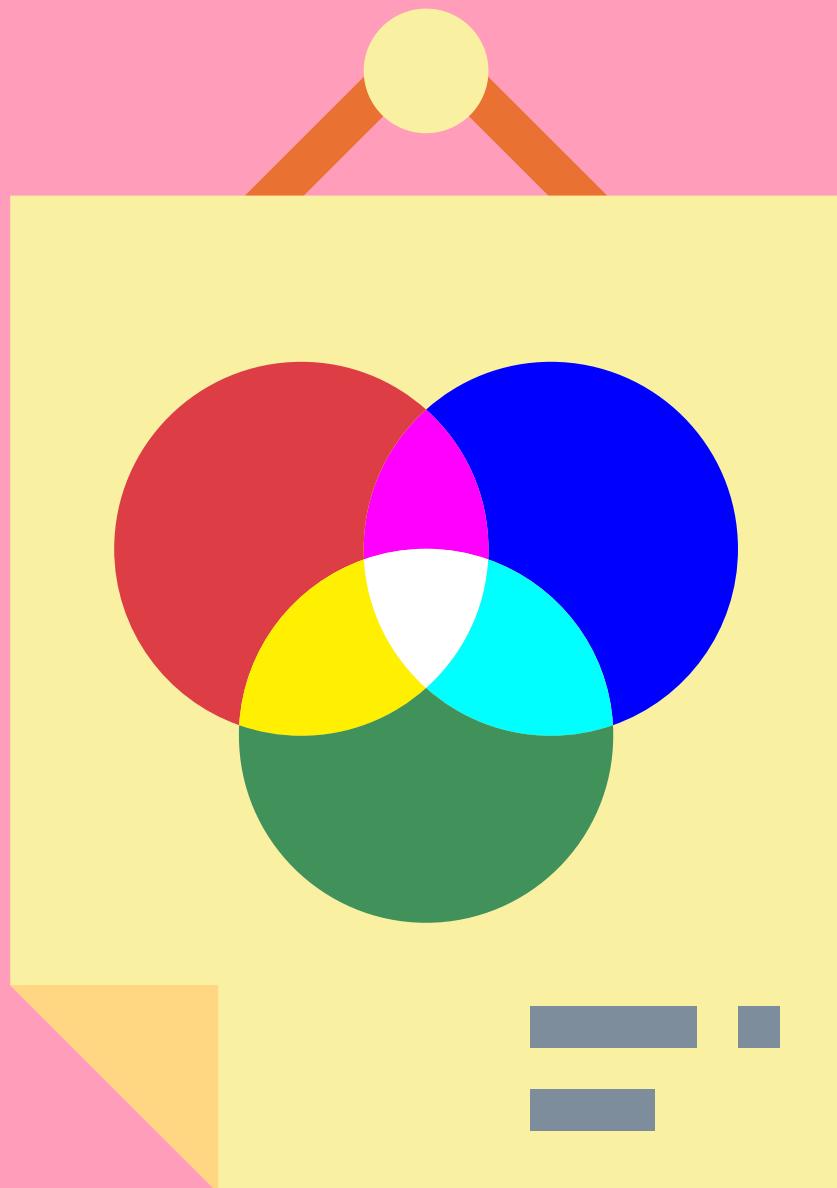
5,6,7,8,9

Hexadecimal

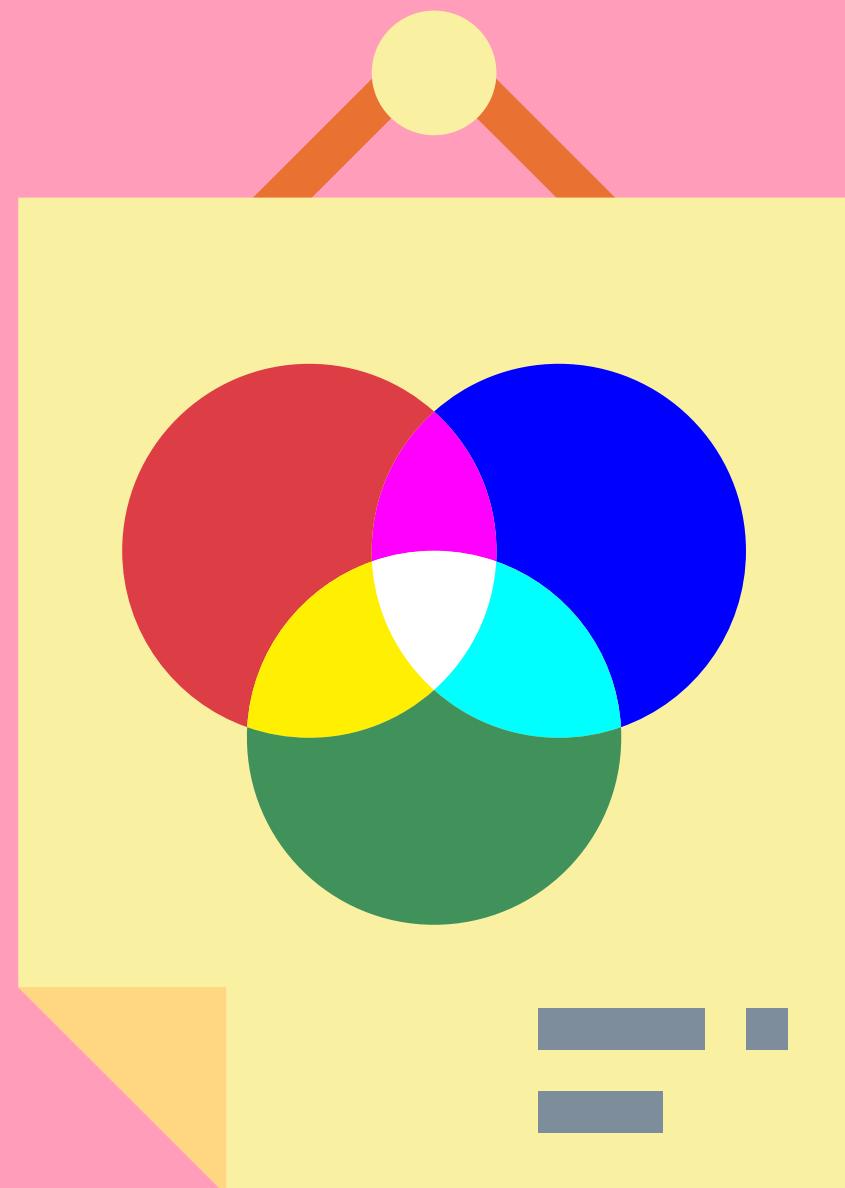
0,1,2,3,4,

5,6,7,8,9,

A,B,C,D,E,F



#fffff00
red green blue



#0f5679

red

green

blue



CSS Text Properties

- **text-align**
- **font-weight**
- **text-decoration**
- **line-height**
- **letter-spacing**



**FONT
SIZE**





Relative

- EM
- REM
- VH
- VW
- %
- AND MORE!

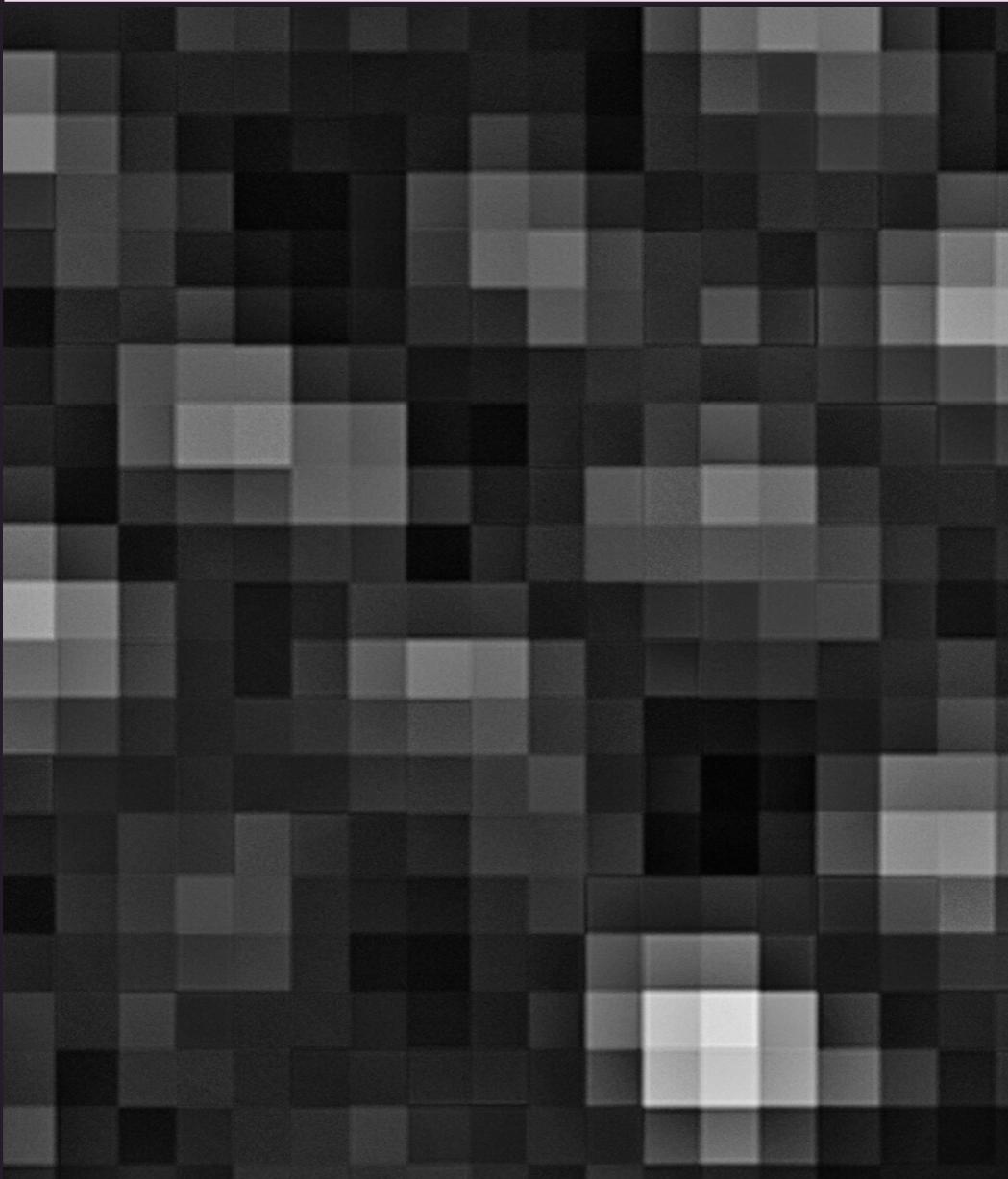
Absolute

- PX
- PT
- CM
- IN
- MM

FONT FAMILY



Absolute Units



**PX - BY FAR THE MOST
COMMONLY USED ABSOLUTE UNIT**

1px does not necessarily equal the width
of exactly one pixel!

Not recommended for responsive websites.

em



EM'S ARE RELATIVE UNITS

With font-size, 1em equals the font-size of the parent. 2em's is twice the font-size of the parent, etc.

With other properties, 1em is equal to the computed font-size of the element itself.

rem



ROOT EMS

Relative to the **root html element's** font-size. Often easier to work with.

If the root font-size is 20px, 1 rem is always 20px, 2rem is always 40px, etc.

CSS SELECTORS

A SUPER IMPORTANT PART OF CSS!

CSS RULES

Everything you do in CSS follows this basic pattern:

```
selector {  
    property: value;  
}
```

CSS RULES

Everything you do in CSS follows this basic pattern:



```
selector {  
  property: value;  
}
```

CSS RULES

Make all `<h1>` elements purple

```
h1 {  
    color: purple;  
}
```

UNIVERSAL SELECTOR

Select everything!

```
* {  
    color: black;  
}
```

ELEMENT SELECTOR

Select all images

```
img {  
    width: 100px;  
    height: 200px;  
}
```

SELECTOR LIST

Select all h1's and h2's

```
h1, h2 {  
  color: magenta;  
}
```

CLASS SELECTOR

Select elements with class of 'complete'

```
.complete {  
    color: green;  
}
```

ID SELECTOR

Select the element with id of 'logout'

```
#logout {  
    color: orange;  
    height: 200px;  
}
```

DESCENDANT SELECTOR

*Select all <a>'s that are nested inside an *

```
li a {  
    color: teal;  
}
```

ATTRIBUTE SELECTOR

Select all input elements where the type attribute is set to "text"



```
input[type="text"] {  
    width: 300px;  
    color: yellow;  
}
```

PSEUDO CLASSES

keyword added to a selector that specifies a special state of the selected element(s)

- :active
- :checked
- :first
- :first-child
- :hover
- :not()
- :nth-child()
- :nth-of-type()

PSEUDO ELEMENTS

Keyword added to a selector that lets you style a particular part of selected element(s)

- ::after
- ::before
- ::first-letter
- ::first-line
- ::selection

What happens when
conflicting styles
target the same
elements?

THE CASCADE

The order your styles are declared in
and linked to matters!



```
h1 {  
  color: red;  
}  
h1 {  
  color: purple;  
}
```

Purple wins!

SPECIFICITY

Specificity is how the browser decides which rules to apply when multiple rules could apply to the same element.

It is a measure of how specific a given selector is. The more specific selector "wins"

ID >

CLASS >

ELEMENT



```
section p {  
    color: teal;  
}
```



ID Selectors



Class,
Attribute, &
Pseudo-Class
Selectors



Element and
Pseudo-Element
Selectors

1

ID Selectors



Class,
Attribute, &
Pseudo-Class
Selectors

```
#submit {  
  color: olive;  
}
```



Element and
Pseudo-Element
Selectors



ID Selectors

1

Class,
Attribute, &
Pseudo-Class
Selectors

2

Element and
Pseudo-Element
Selectors

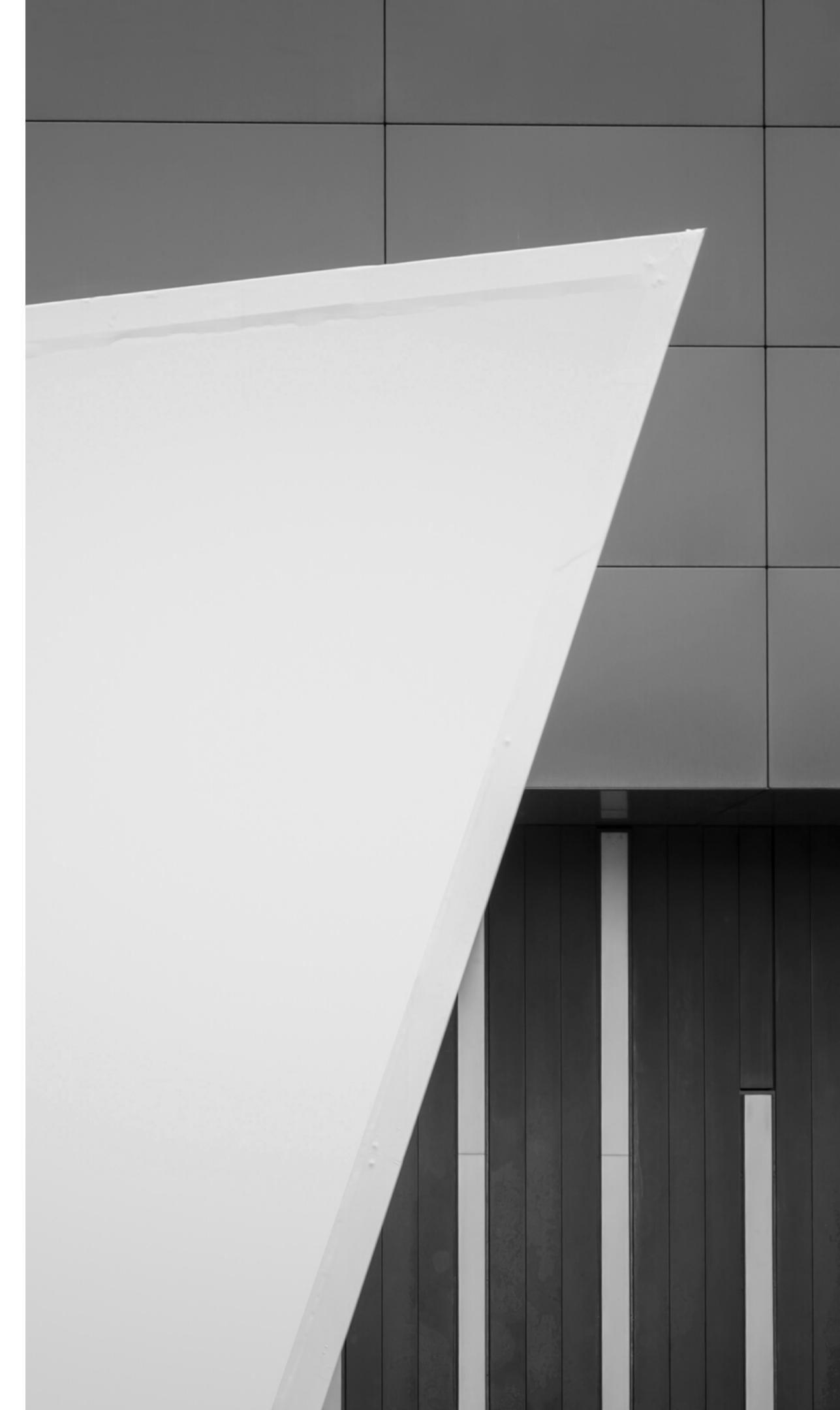
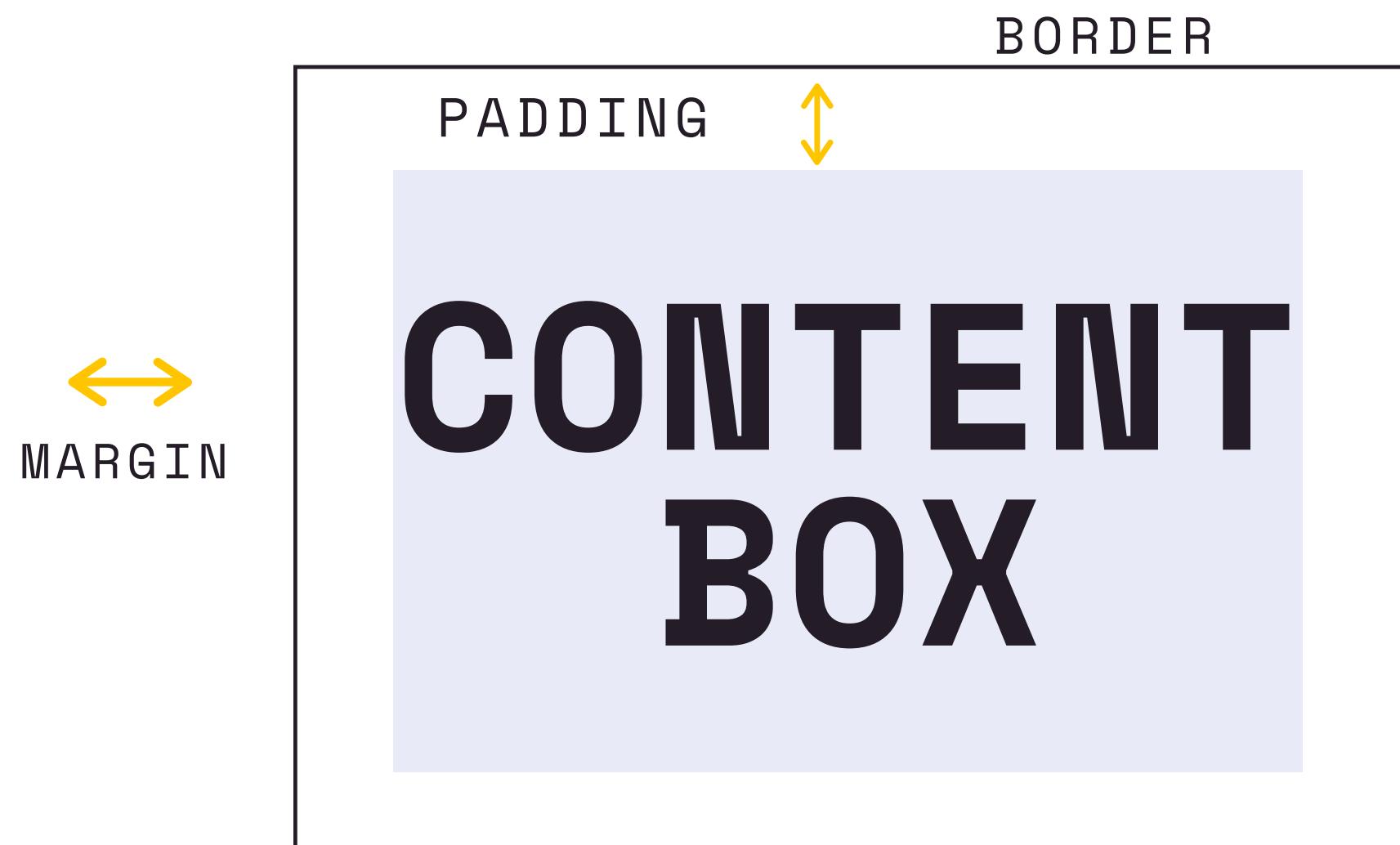


```
nav a.active {  
    color: orange;  
}
```

THE CSS BOX MODEL

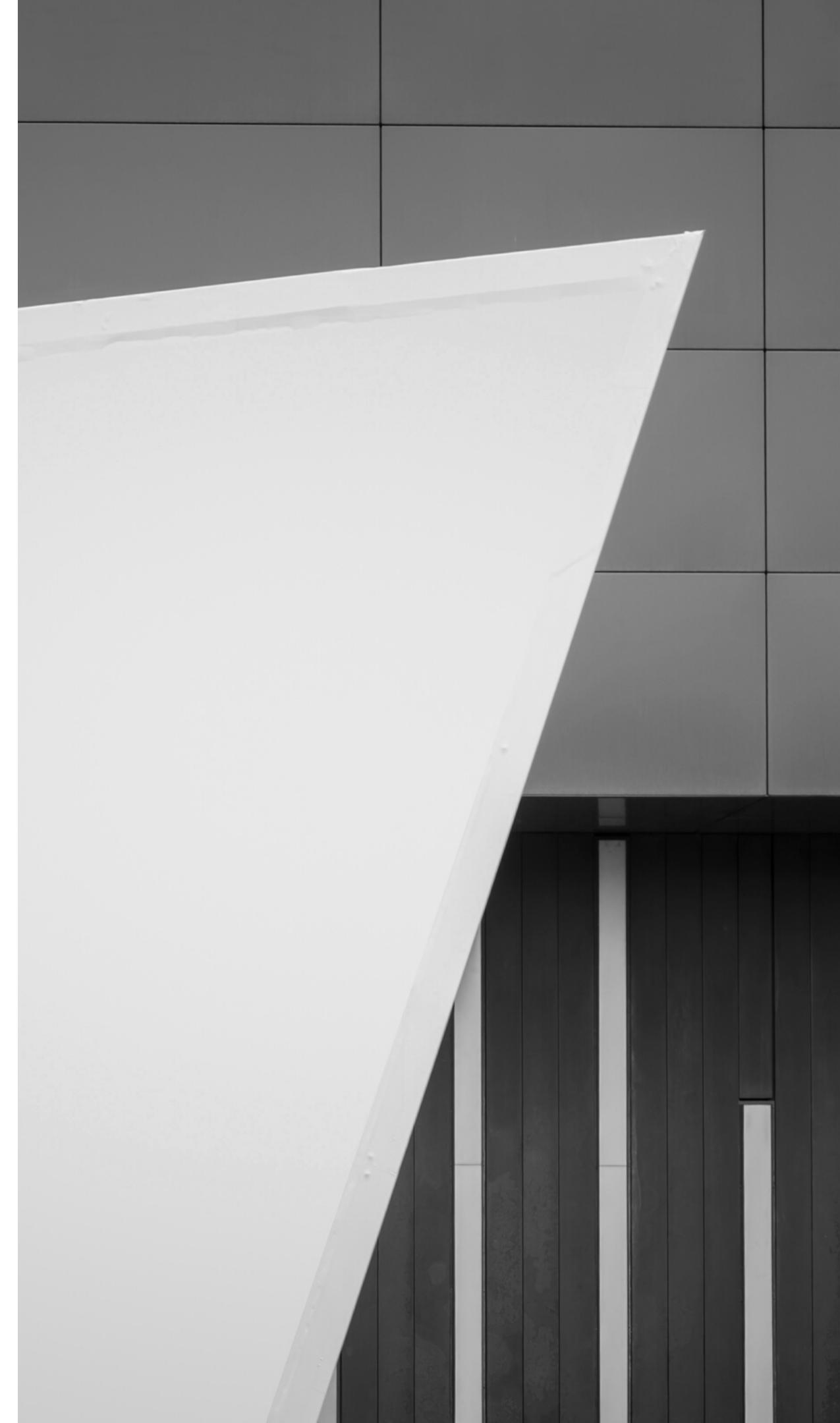
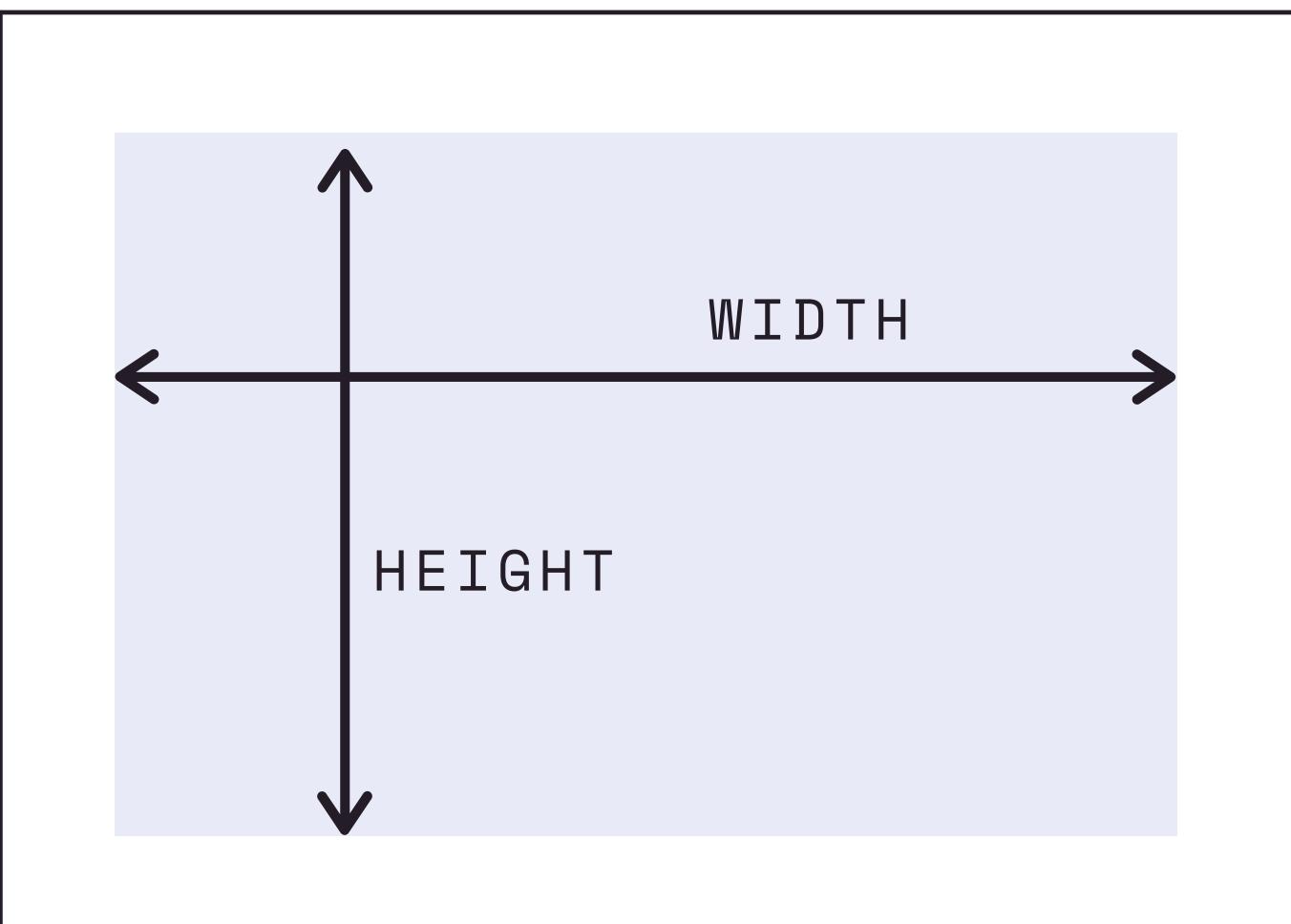


The Box Model



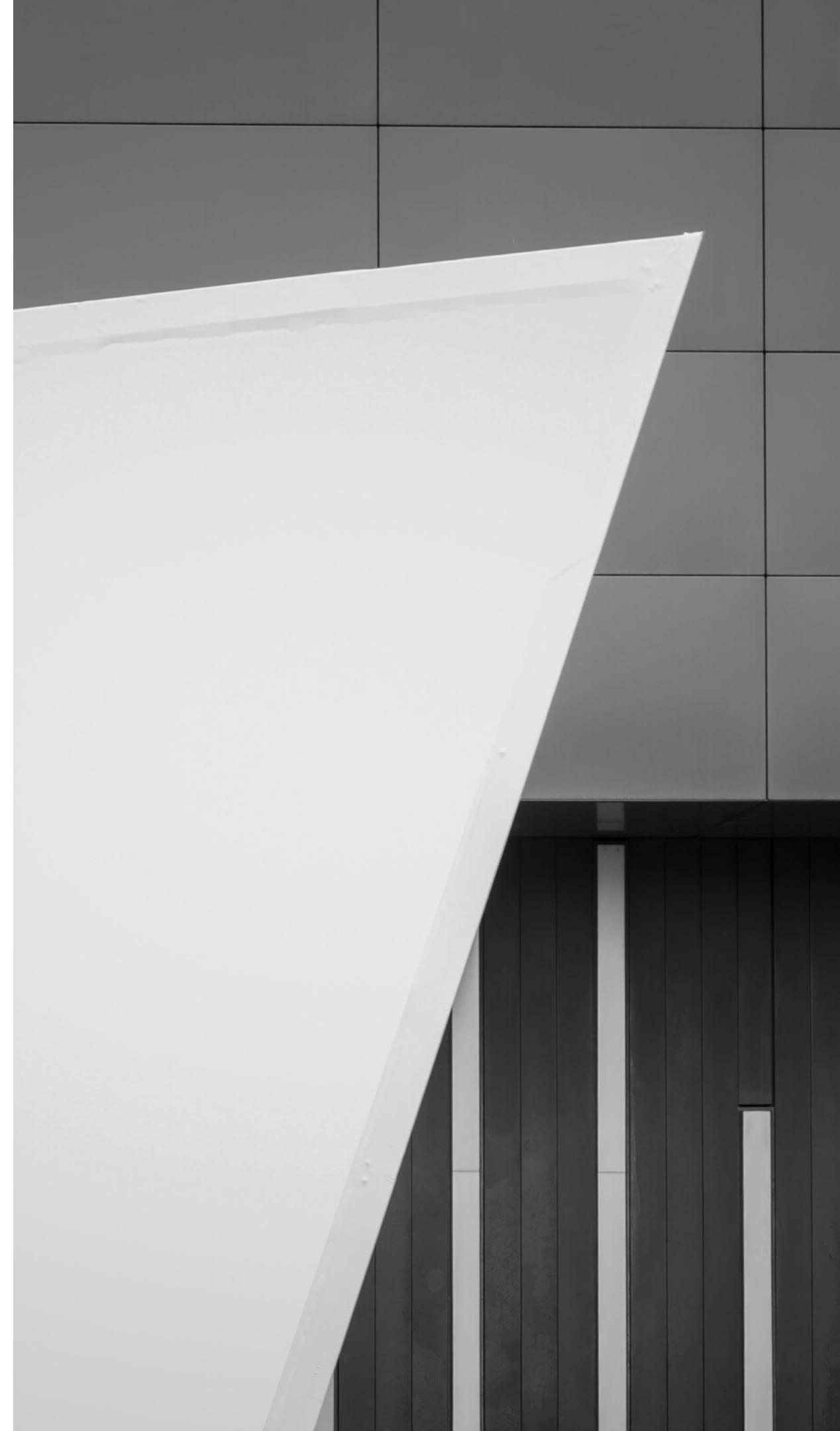
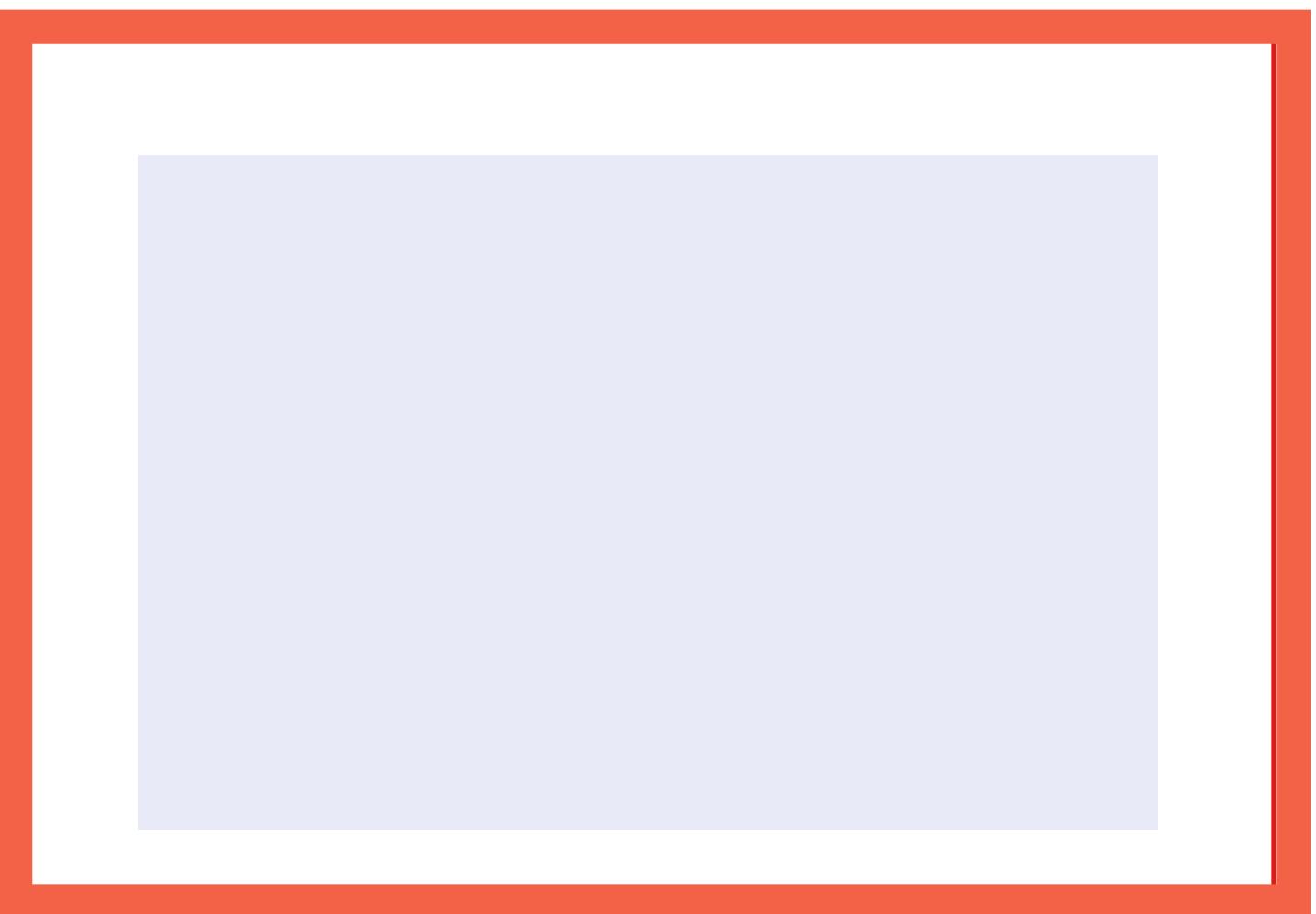


Width & Height





Border



Border Properties

(the important ones)

BORDER-WIDTH

Controls the thickness of the border.

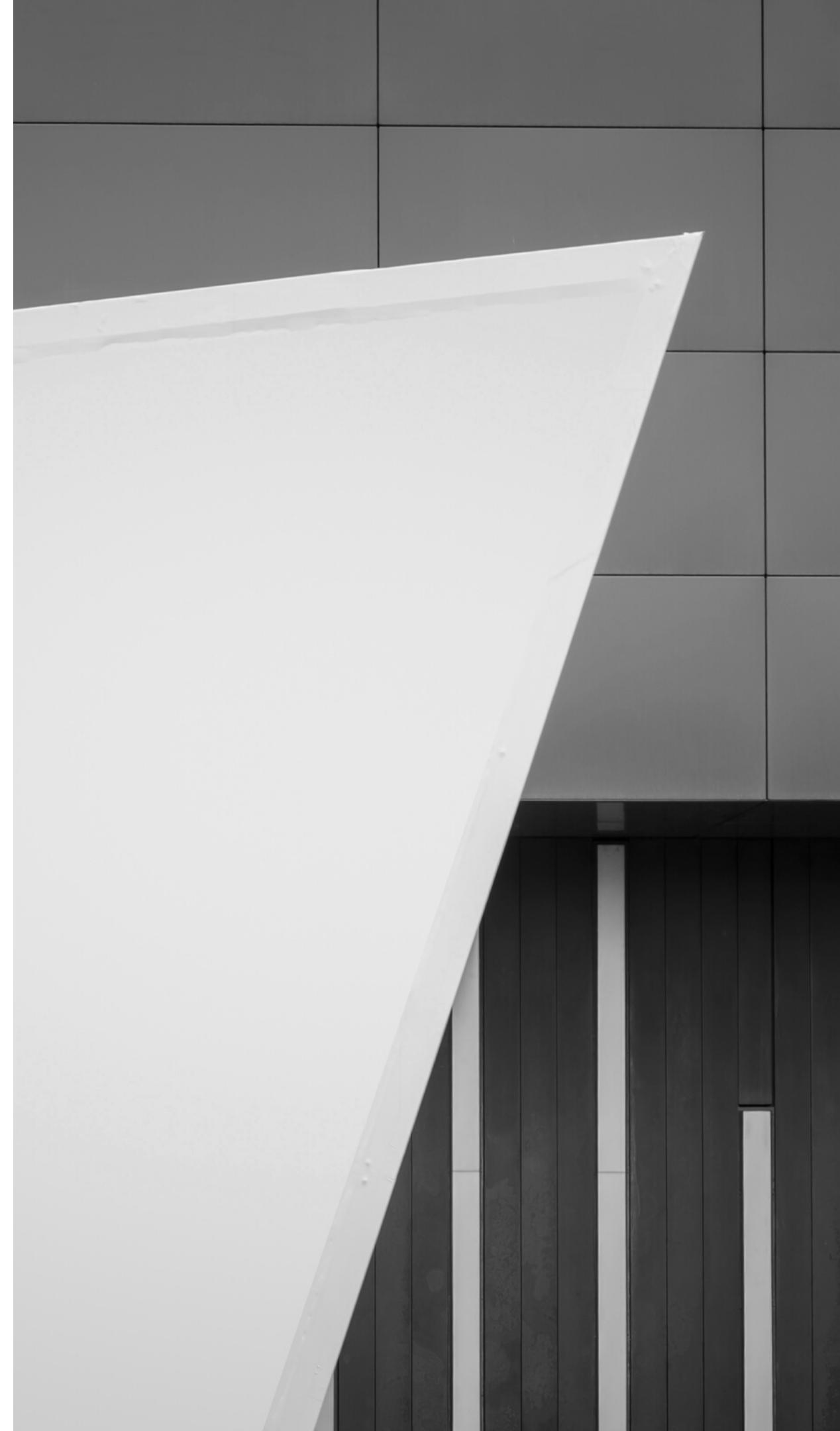
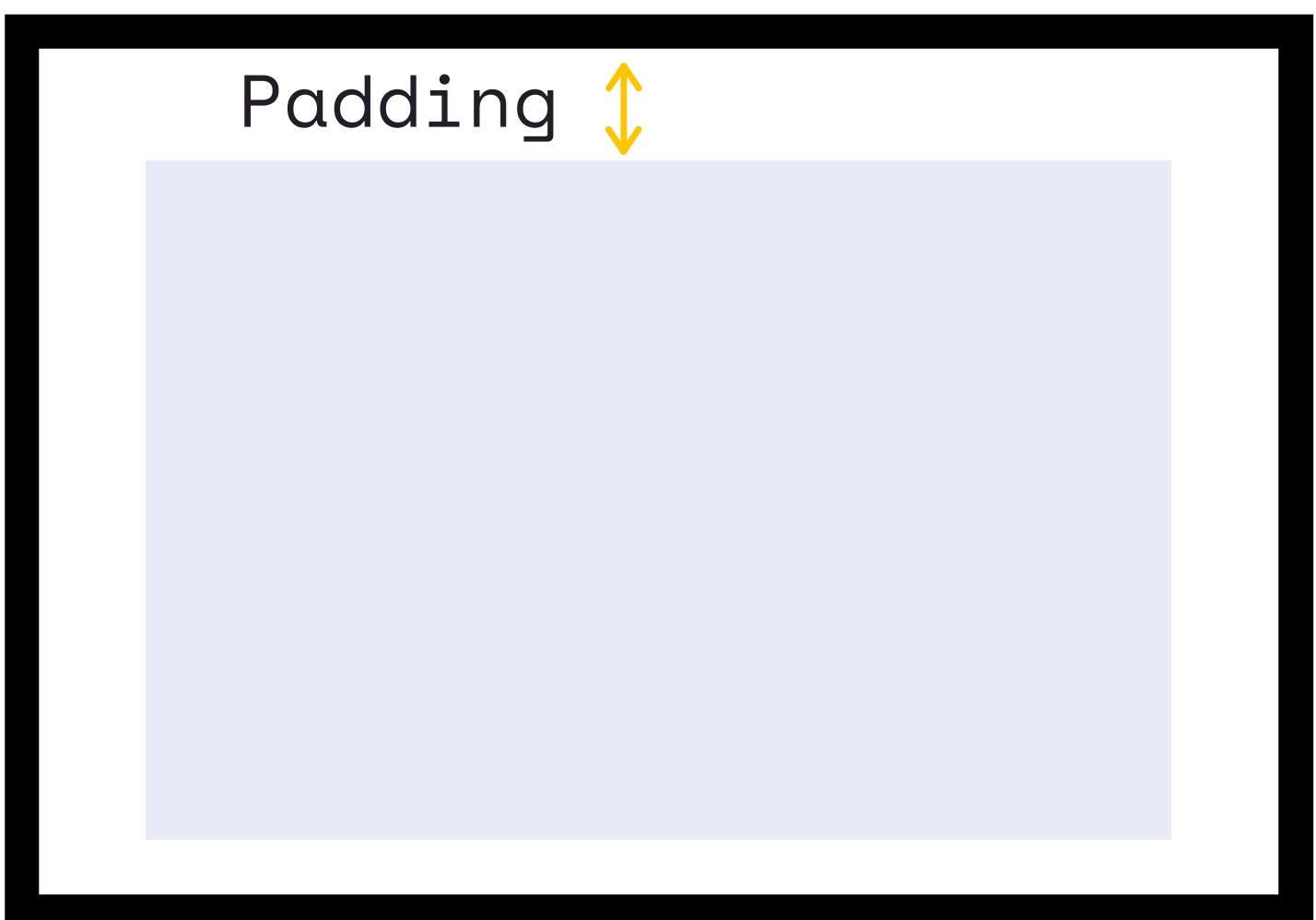
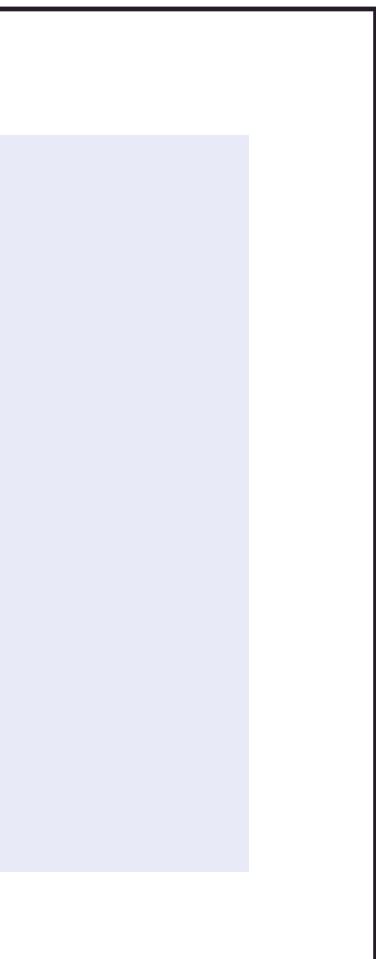
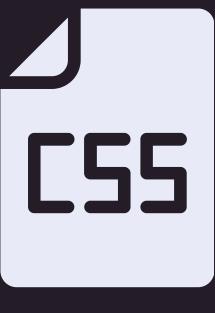
BORDER-COLOR

Controls the...color of the border

BORDER-STYLE

Controls the line style - dashed, solid, etc.





```
="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-sca  
ef="/favicon.ico" type="image/x-icon">  
icon.ico" type="image/x-icon">  
  
'stylesheet' href="css/materialize.min.css" media="screen, print">  
f="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/  
ef="/css/animate.css">  
ef="/css/theme.css">  
  
rapper">  
tainer">  
class="brand-logo hide-on-med-and-up"><span style="font-size:  
  
ainer">  
ow">  
12 m8 l8 hide-on-small-only">  
ight:400">starter</span><span style="font-size:  
stup network and workspace tool</span>  
span 10="banner-rear">  
nner">
```

Individual Properties

PADDING - LEFT
PADDING - RIGHT
PADDING - BOTTOM
PADDING - TOP

Shorthand Property

SET ALL FOUR SIDES AT ONCE !



Padding Shorthand Property

Apply to all four sides

`padding: 10px;`

vertical | horizontal

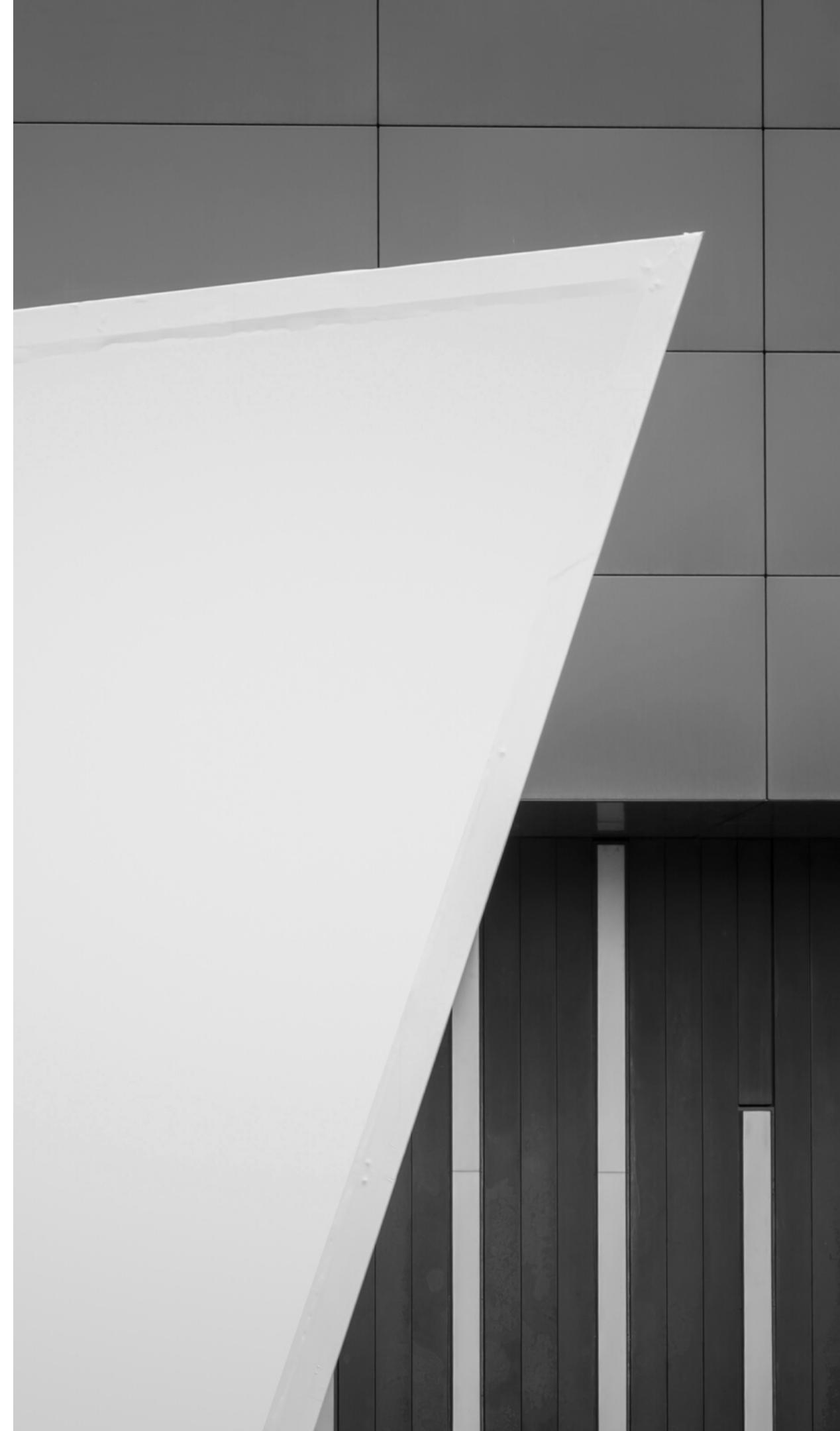
`padding: 5px 10px;`

top | horizontal | bottom

`padding: 1px 2px 2px;`

top | right | bottom | left

`padding: 5px 1px 0 2px;`



Individual Properties

MARGIN-LEFT

MARGIN-RIGHT

MARGIN-BOTTOM

MARGIN-TOP

Shorthand Property

SET ALL FOUR
SIDES AT
ONCE !



Margin Shorthand Property

Apply to all four sides

`margin: 10px;`

vertical | horizontal

`margin: 5px 10px;`

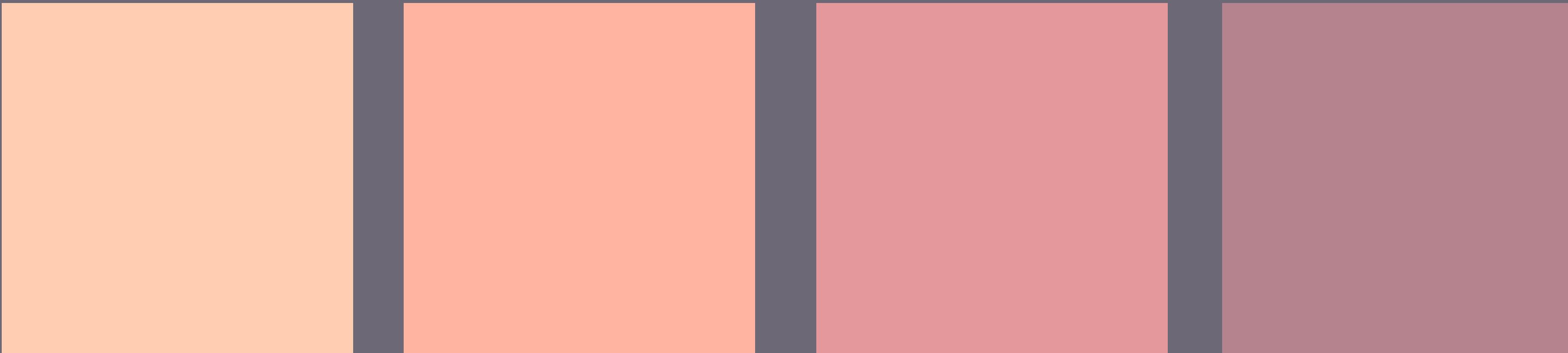
top | horizontal | bottom

`margin: 1px 2px 2px;`

top | right | bottom | left

`margin: 5px 1px 0 2px;`

Display Property



INLINE ELEMENTS



- Inline elements fit in alongside other elements
- Block level elements take up a whole "block" of space.

BLOCK ELEMENTS



Display Property

(our first encounter)

INLINE

Width & Height are ignored. Margin & padding push elements away horizontally but not vertically.

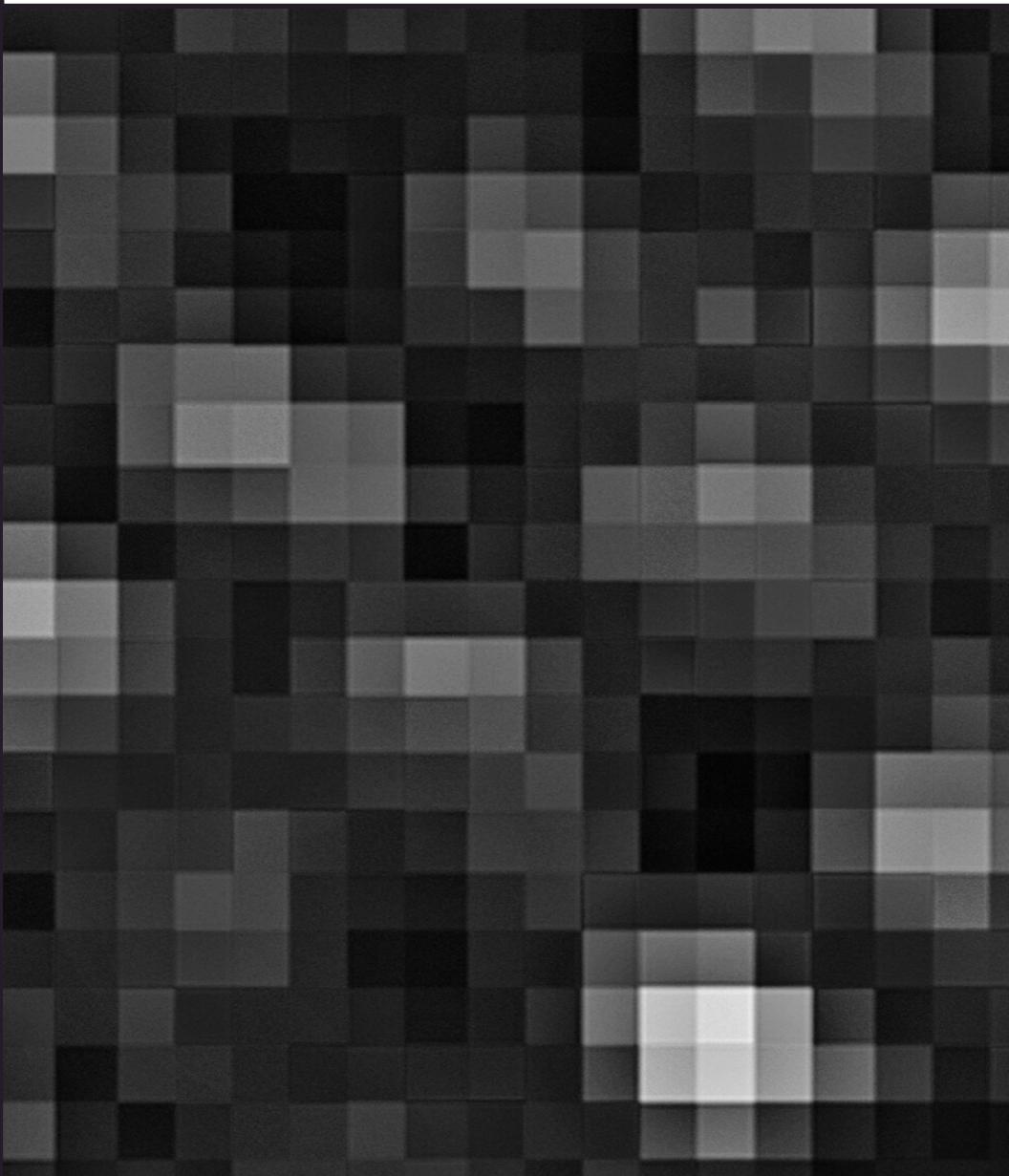
BLOCK

Block elements break the flow of a document. Width, Height, Margin, & Padding are respected.

INLINE-BLOCK

Behaved like an inline element except Width, Height, Margin, & Padding are respected

Absolute Units



**PX - BY FAR THE MOST
COMMONLY USED ABSOLUTE UNIT**

1px does not necessarily equal the width
of exactly one pixel!

Not recommended for responsive websites.

percentages



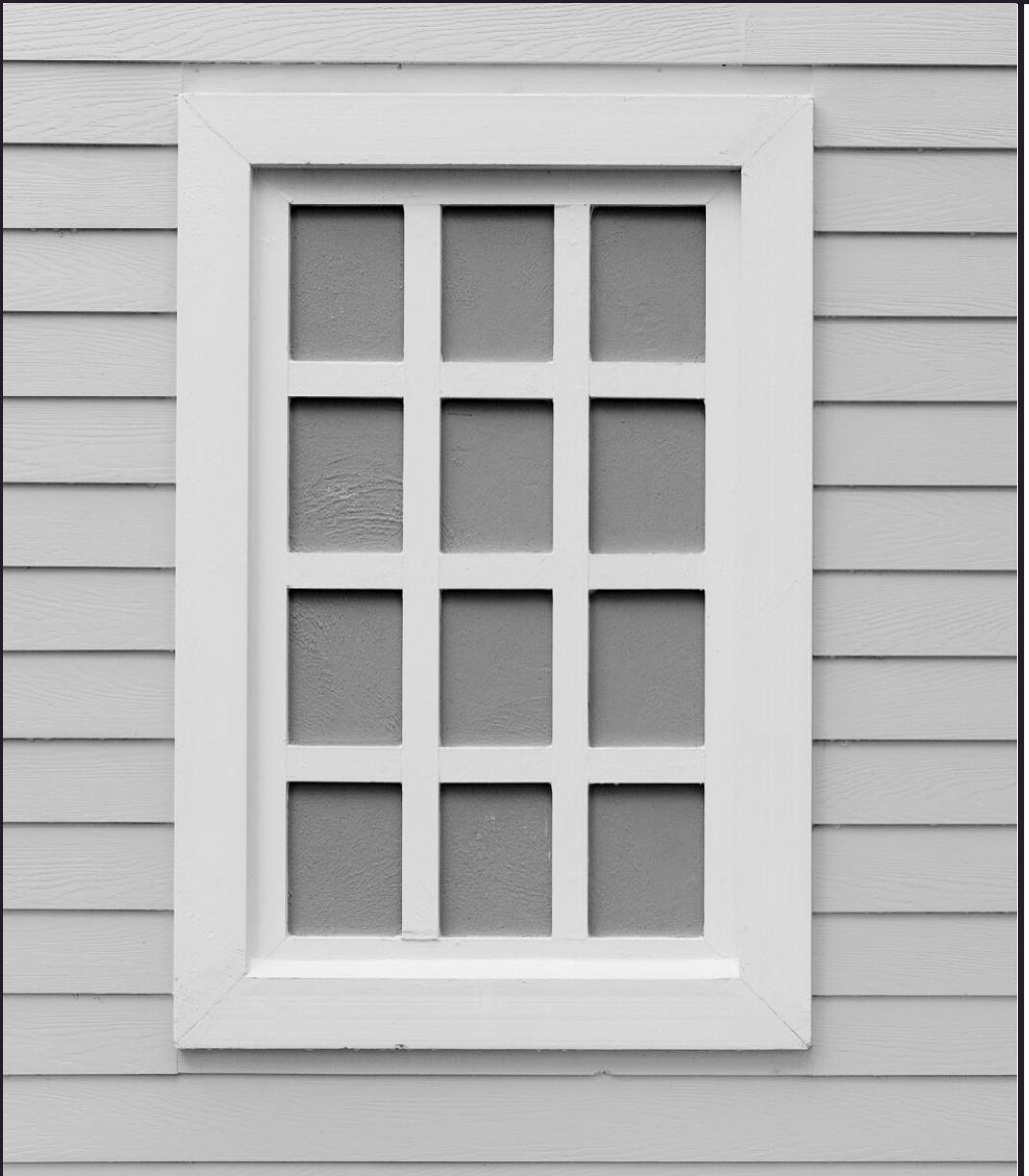
**PERCENTAGES ARE ALWAYS
RELATIVE TO SOME OTHER VALUE**

Sometimes, it's a value from the parent
and other times it's a value from the
element itself.

width: 50% - half the width of the parent

line-height: 50% - half the font-size of
the element itself

vw & vh



VIEW HEIGHT & VIEW WIDTH

1vw is 1% of the width of the viewport

1vh is 1% of the height of the viewport

For example: height: 100vh would make an element take up the full height on screen



Opacity +
Alpha Channel

rgba(0, 209, 112, 0.5)

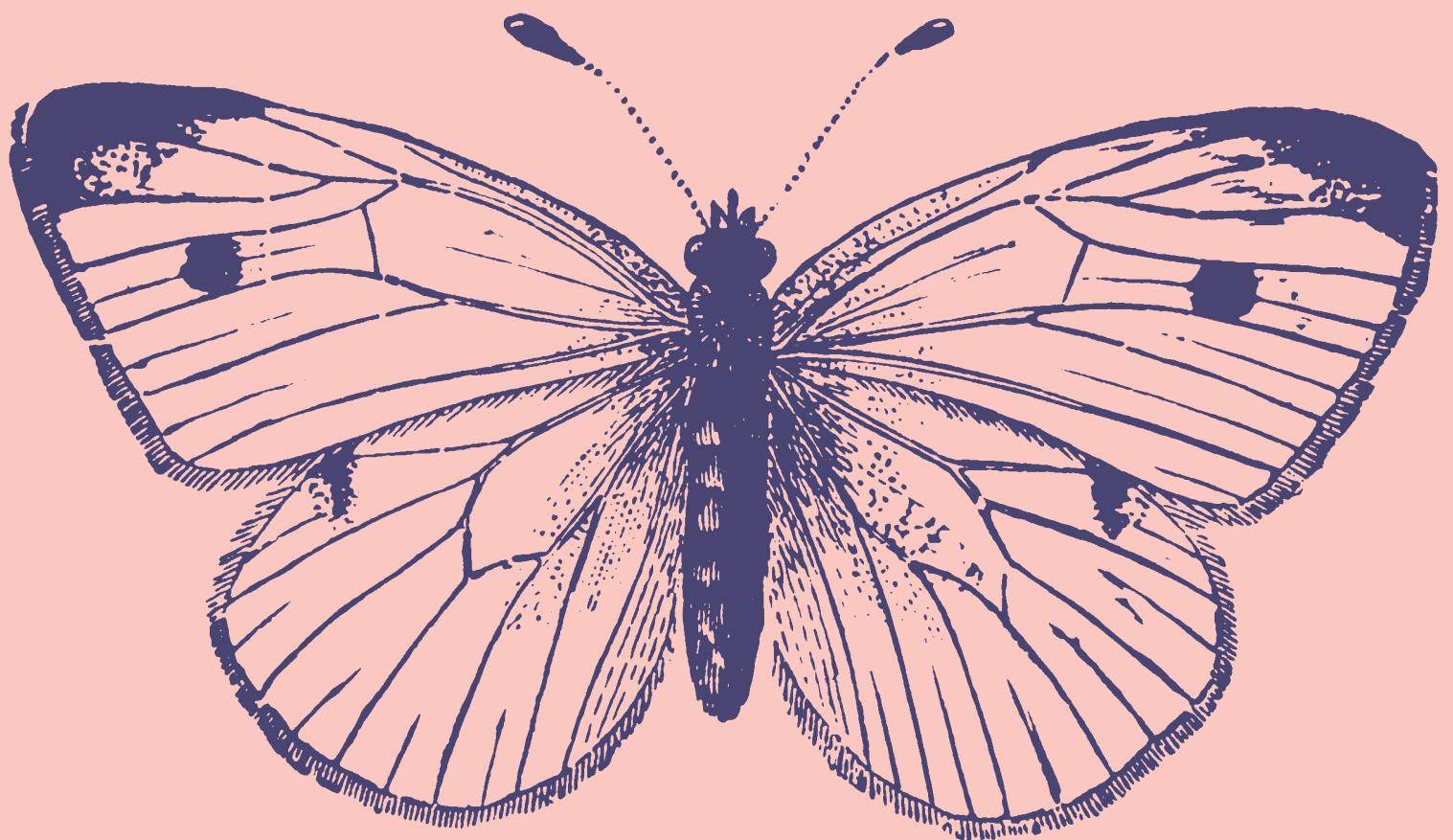
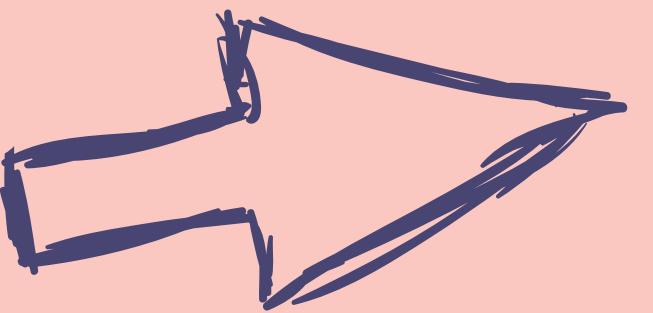
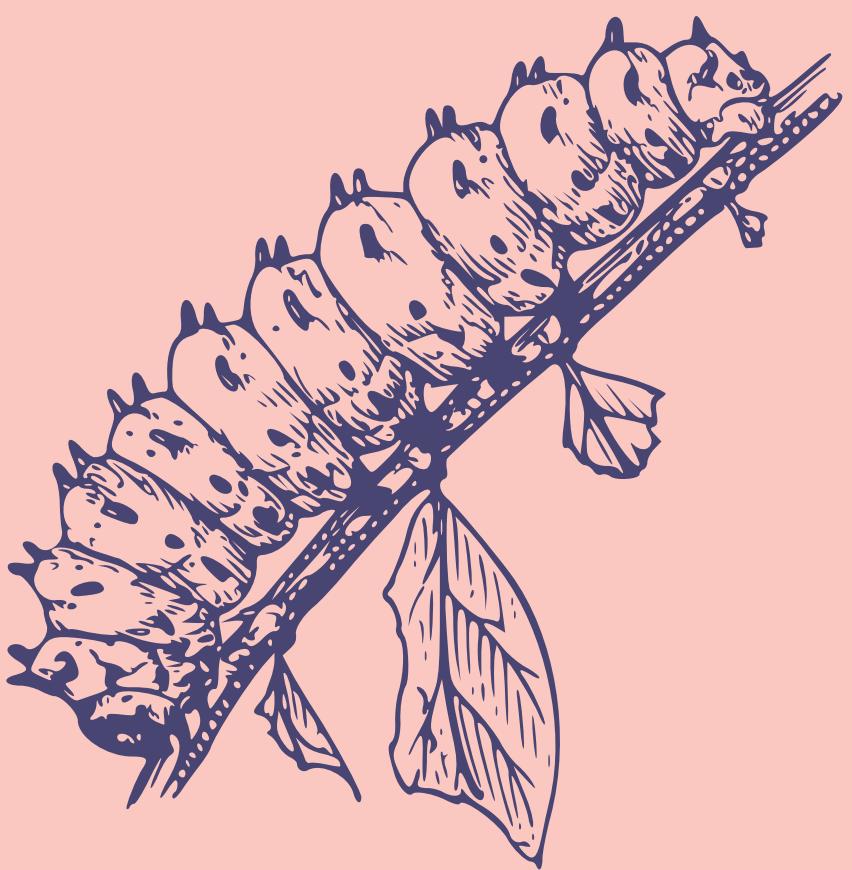
red

green

blue

alpha

Transitions



Transition:

PROPERTY NAME | DURATION | TIMING FUNCTION | DELAY

Transform





```
/* Function values */
transform: matrix(1.0, 2.0, 3.0, 4.0, 5.0, 6.0);
transform: matrix3d(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1);
transform: perspective(17px);
transform: rotate(0.5turn);
transform: rotate3d(1, 2.0, 3.0, 10deg);
transform: rotateX(10deg);
transform: rotateY(10deg);
transform: rotateZ(10deg);
transform: translate(12px, 50%);
transform: translate3d(12px, 50%, 3em);
transform: translateX(2em);
transform: translateY(3in);
transform: translateZ(2px);
transform: scale(2, 0.5);
transform: scale3d(2.5, 1.2, 0.3);
transform: scaleX(2);
transform: scaleY(0.5);
transform: scaleZ(0.3);
transform: skew(30deg, 20deg);
transform: skewX(30deg);
transform: skewY(1.07rad);

/* Multiple function values */
transform: translateX(10px) rotate(10deg) translateY(5px);
```



background

GOOGLE FONTS

GOOGLE FONTS