

SSDP 1100 – Day 6

Course: Front-End Web Development Essentials
Instructor: Gabbie Bade

Agenda

- CSS Pseudo-elements
- CSS Positioning
- Assignment #4

Pseudo Elements

CSS Pseudo Elements

A pseudo-element is a part of an existing HTML element that can be selected and styled.

For example: the first letter or the first line of some text.

Two common pseudo-elements are **::before** and **::after**. They act like a new element was added to the HTML.

<https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements>

Pseudo Element Syntax

The pseudo-element selector must first select an element by its type, class, or ID. Then it can target the pseudo-element.

Double colon indicates
a pseudo-element

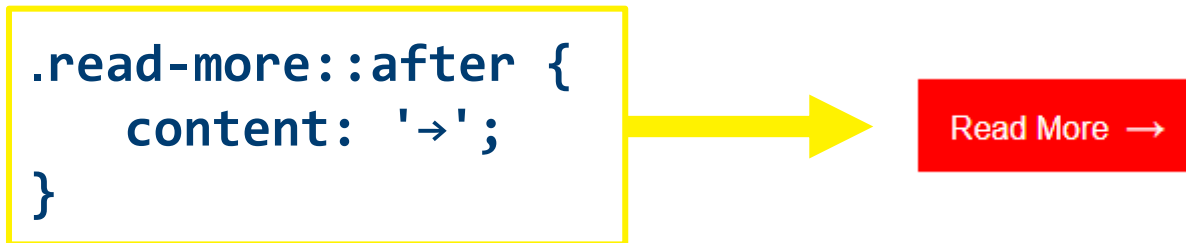


```
p::first-letter {  
  font-size: 2rem;  
  color: red;  
}
```

Pseudo Elements for Design

Pseudo-elements should **not** be used for content and only to add to the design.

The content should make sense without the pseudo-element.



Selectors vs Elements

Pseudo-elements are used to style part of an element.

Example: the first letter of a paragraph.

Pseudo-classes are used to style an element based on its state.

Example: the hover state of a button.

CSS Positioning

CSS Position Property

The CSS `position` property sets how an element is positioned in a document.

The `top`, `right`, `bottom`, `left` properties determine the final location of positioned elements on the x and y axis.

The `z-index` property determines the final location of the positioned elements on the z axis.

<https://developer.mozilla.org/en-US/docs/Web/CSS/position>

Use Position Sparingly

Do **not** overuse the CSS `position` property.

Flexbox and Grid should be your default ways to handle layout.

Positioning is used for specific cases when standard methods of layout do not work.

CSS Position Values

The position property has five possible values:

- `static`
- `relative`
- `absolute`
- `fixed`
- `sticky`

Related Properties

In addition to setting position, you may also use these CSS properties but they are not required:

- top
- bottom
- left
- right
- z-index

Position – Static

This is the default value of the position property for every HTML element and means the element will appear where it is in the HTML document.

The top, bottom, left, right, z-index properties are ignored if set on an element with `position: static`.

If you ever use this, it will only be to override another position value.

Position – Relative

Relative is similar to Static except you can now use top, bottom, left, right, z-index.

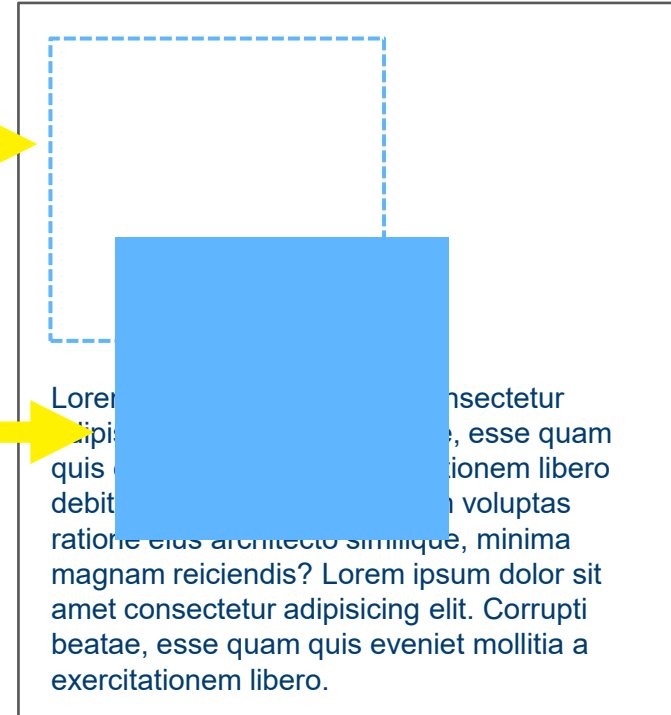
The element is positioned normally then offset **relative to itself** based on the values of top, right, bottom, and left.

Moving the element does not impact elements around them and the original space for the element remains.

Position – Relative

Original location of the element

```
.box {
    position: relative;
    top: 20rem;
    left: 2rem;
}
```



Position – Relative

You generally use `position: relative` so you can use the remaining positions **inside** of a container.

In particular, `position: absolute` works well when the parent is relatively positioned.

Position – Absolute

Absolute is similar to relative except it removes the element from the normal flow so the original space is removed.

If no parent element has a position set other than static, then the element will position based on the browser window when using top, bottom, left, right, z-index.

Position – Absolute

Original location of the element.
The text has moved up to fill that empty space.

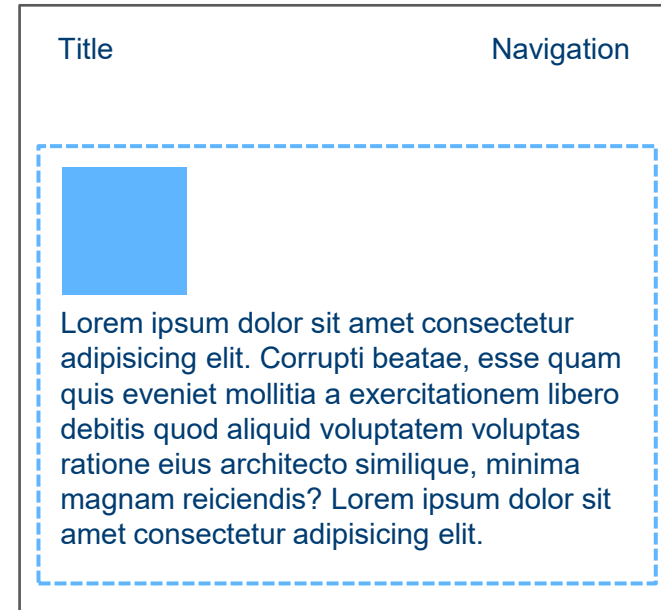
```
.box {  
  position: absolute;  
  top: 20rem;  
  left: 2rem;  
}
```

Lorem ipsum dolor sit amet consectetur
adipiscing elit. Corrupti beatae, esse quam
quis eveniet mollitia a exercitationem libero
debitis quod aliquid voluptatem voluptas
ratione eius architecto similique, minima
magnam reiciendis? Lorem ipsum dolor sit
amet elit. Corrupti
beatae et mollitia a
exercitationem

Position – Absolute

An example of a page with no positioning set.

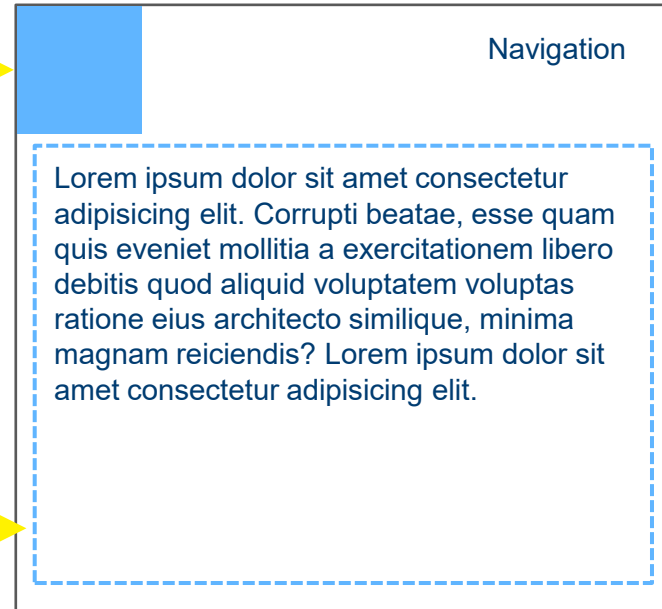
This is simply the default layout.



Position – Absolute

```
.box {  
  position: absolute;  
  top: 0;  
  left: 0;  
}
```

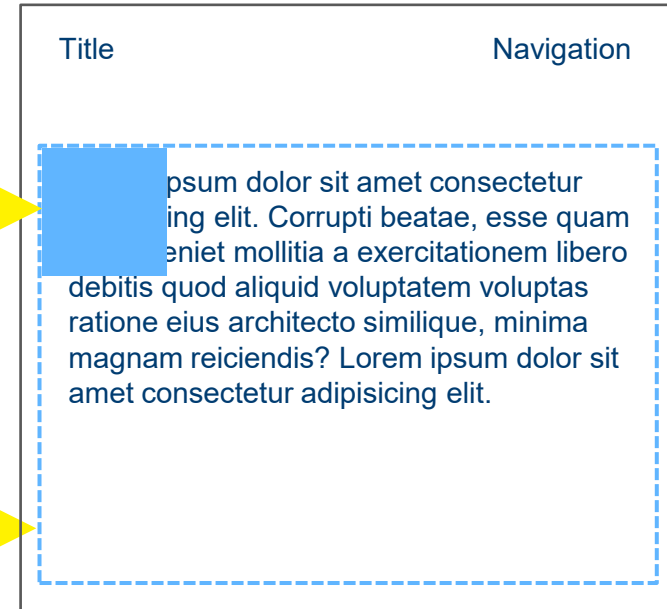
Parent element has no position
set so defaults to static.



Position – Absolute

```
.box {  
  position: absolute;  
  top: 0;  
  left: 0;  
}
```

```
.parent {  
  position: relative;  
}
```



Position – Fixed

Fixed is similar to absolute except it is **always** relative to the viewport.

Using the top, bottom, left, right, z-index properties is based on the viewport, not the webpage.

For example, setting bottom: 0; will place the element at the bottom of the viewable browser window, not the bottom of the webpage.

Position – Fixed

```
.box {  
  position: fixed;  
  top: 0;  
  left: 0;  
}
```



amet consectetur
eatae, esse quam
quis eveniet mollitia a exercitationem libero
debitis quod aliquid voluptatem voluptas
ratione eius architecto similique, minima
magnam reiciendis? Lorem ipsum dolor sit
amet consectetur adipisicing elit.

Position – Fixed

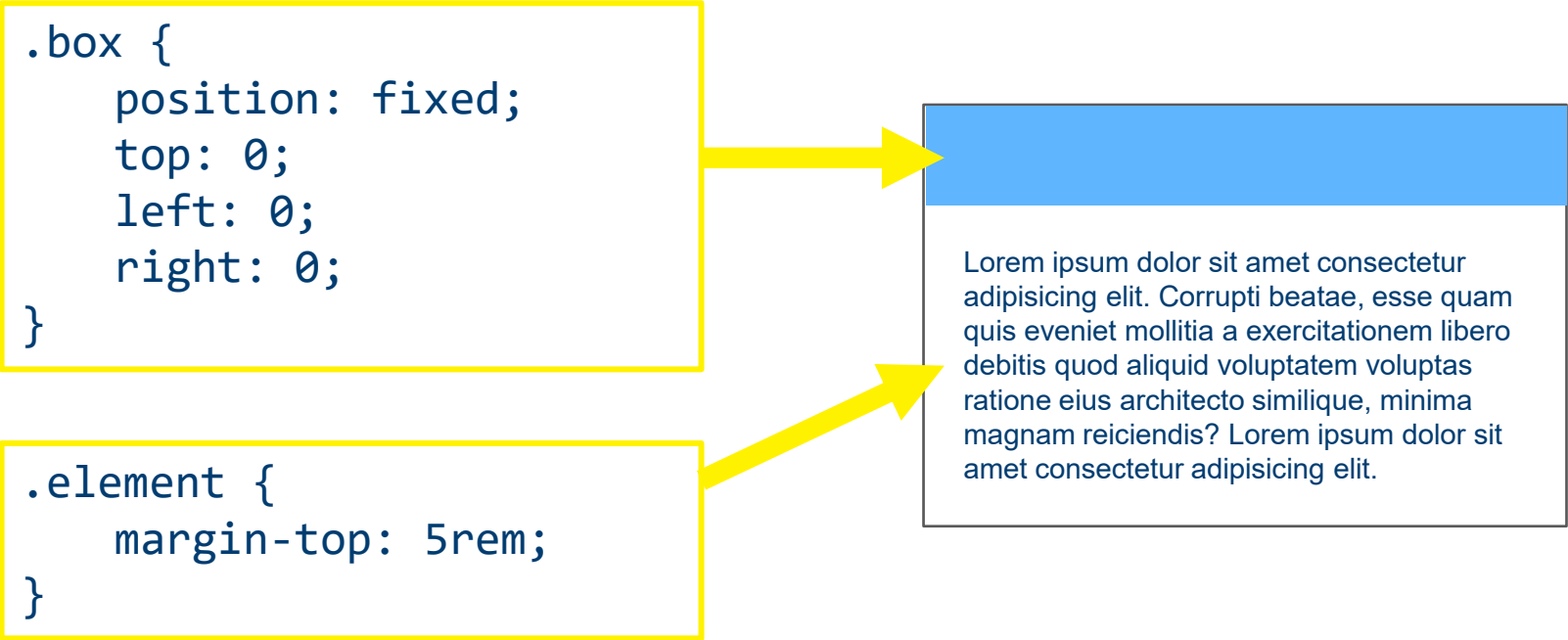
When you set an element to `position: fixed`, make sure you consider that some elements may have flown behind it.

You will often want to offset other elements by the height or width of the fixed element.

Position – Fixed

```
.box {  
  position: fixed;  
  top: 0;  
  left: 0;  
  right: 0;  
}
```

```
.element {  
  margin-top: 5rem;  
}
```



The diagram illustrates the application of CSS rules to a box. On the left, two CSS rule blocks are shown. The first rule, `.box { position: fixed; top: 0; left: 0; right: 0; }`, is connected by a yellow arrow to the top portion of a box on the right, which has a blue header. The second rule, `.element { margin-top: 5rem; }`, is connected by another yellow arrow to the main content area of the box. This content area contains a paragraph of Lorem Ipsum text.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Corrupti beatae, esse quam quis eveniet mollitia a exercitationem libero debitis quod aliquid voluptatem voluptas ratione eius architecto similique, minima magnam reiciendis? Lorem ipsum dolor sit amet consectetur adipisicing elit.

Position – Sticky

Sticky is a hybrid of relative and fixed. It must be used with at least one of top, bottom, left, right otherwise it will be the same as relative.

It functions as relative to start until the viewport reaches the position defined by one of those four properties, then the element functions similar to fixed but stays within its parent container.

Position – Sticky

Check the example provided by MDN to see it in action...

https://developer.mozilla.org/en-US/docs/Web/CSS/position#Result_3

Z-Index

The z-index property can be used on any element with a position set other than static to determine how items stack on one another visually.

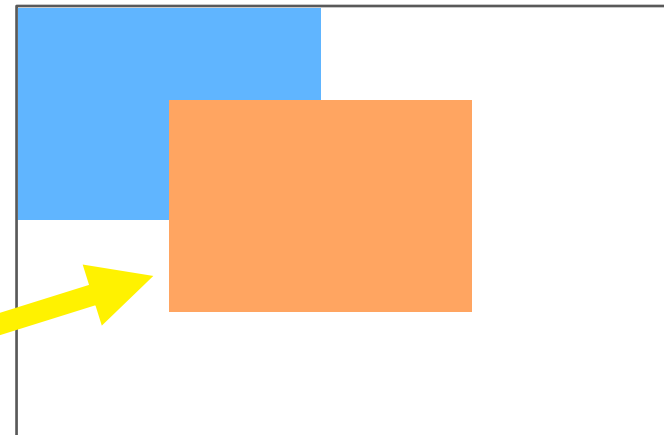
It accepts positive or negative number values with higher values stacking over smaller values.

<https://developer.mozilla.org/en-US/docs/Web/CSS/z-index>

Z-Index

```
.box-1 {  
  position: absolute;  
  top: 0;  
  left: 0;  
  z-index: 1;  
}
```

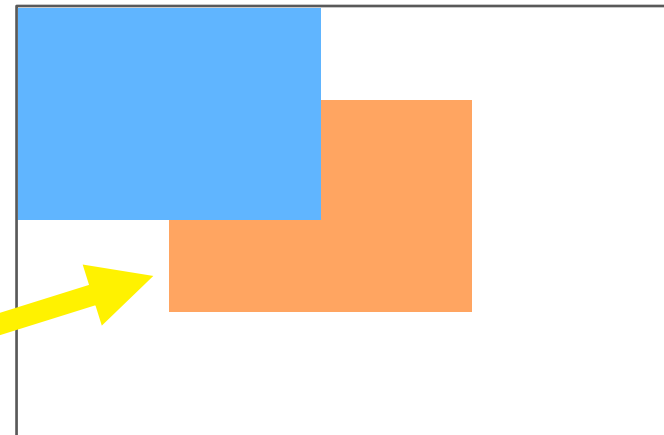
```
.box-2 {  
  position: relative;  
  z-index: 2;  
}
```



Z-Index

```
.box-1 {  
  position: absolute;  
  top: 0;  
  left: 0;  
  z-index: 2;  
}
```

```
.box-2 {  
  position: relative;  
  z-index: 1;  
}
```



Scroll-padding

The **scroll-padding** property can be used to prevent content appearing behind position: fixed or position: sticky elements when clicking in-page links.

We can set **scroll-padding-top** on the HTML element of our FAQ page to address this issue.

<https://developer.mozilla.org/en-US/docs/Web/CSS/scroll-padding>

Scroll-behavior

The **scroll-behavior** property can be used to make the browser either jump to an element or smooth scroll to it when clicking an in-page link.

This property has two possible values: **auto** or **smooth**.

<https://developer.mozilla.org/en-US/docs/Web/CSS/scroll-behavior>

Text over Images

Read this two-part article about designing accessible text over images.

It has tons of great examples worth looking at for inspiration.

<https://www.smashingmagazine.com/2023/08/designing-accessible-text-over-images-part1/>

<https://www.smashingmagazine.com/2023/08/designing-accessible-text-over-images-part2/>

Assignment #4

Assignment #4

- Please refer to Assignment #4 in the Learning Hub.
- To submit the assignment, you can do **one** of these:
 - Have me check your assignment in class before 4pm.
 - Zip today's **folder** and upload it to the Learning Hub before next class.
- If you have questions or need guidance, just ask!

Resources

CSS Position (MDN)

<https://developer.mozilla.org/en-US/docs/Web/CSS/position>

CSS Position (CSS-Tricks)

<https://css-tricks.com/almanac/properties/p/position/>

QUESTIONS & ANSWERS