

CENG 3507

Web Development & Programming

Week X

Course Topic

Instructor
Bekir Taner Dinçer

Teaching Assistant
Selahattin Aksoy



MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ
COMPUTER ENGINEERING
2024 – Fall

1

Today's schedule

Today

- position
- Random helpful CSS
 - calc, variables, background properties
- Mobile layouts
 - em and rem



MSKU CENG

CENG-3507 Web Development & Programming

2

Question of the day!

"I can think of 12 different ways to implement this!
Which one is best?"



MSKU CENG

CENG-3507 Web Development & Programming

3

Simplicity above all else

Always prefer **simplicity**.

Other tips:

- **Separation of concerns:** HTML should contain content NOT style, CSS should contain style NOT content
- **Descriptive HTML tags:** Make your HTML more readable by using e.g. `<header>` instead of `<div>` when appropriate
- **Reduce redundancy:** Try grouping styles, using descendant selectors to reduce redundancy (see past slides for details)



MSKU CENG

CENG-3507 Web Development & Programming

4

Last time

More flexbox stuff:

- `flex-shrink`
 - Default value is 1 (**on** by default)
 - This CSS property is on the flex *item*, not container
- `flex-grow`
 - Default value is 0 (**off** by default)
 - This CSS property is on the flex *item*, not container

vh / vw / box-sizing



5

From Previous Lecture

How do we create this look?

<https://codepen.io/pen?template=BaVBPag>

Home

About

News

Links

Contact

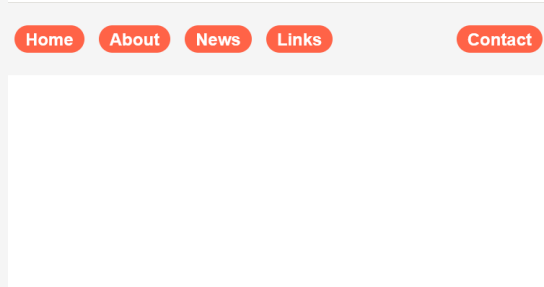


6

From Previous Lecture

```
HTML
<body>
  <nav>
    <a>Home</a>
    <a>About</a>
    <a>News</a>
    <a>Links</a>
    <a id="split">Contact</a>
  </nav>
</body>

CSS
#split {
  margin-left: auto;
}
```



MSKU CENG

CENG-3507 Web Development & Programming

7

Position

Another rendering mode



MSKU CENG

CENG-3507 Web Development & Programming

8

Moving things with position

Positioned layout lets you define precisely where an element should be in the page ([mdn](#)).

You can use positioned layout doing the following:

1. Define a **position** method:
static, fixed, absolute, relative
2. Define **offsets**: top, left, bottom, and right
3. (optional) Define **z-index** for overlapping layers ([mdn](#))



MSKU CENG

CENG-3507 Web Development & Programming

9

position: static (nothing happens!)

- static is the default value for position
- If you use top / left / bottom / right without setting a non-static position, nothing will happen

```
<body>
  <h1>Puppy</h1>
  <p>A puppy is a juvenile dog. Some puppies
  <h2>Development</h2>
  <p>At first, puppies spend the large major
  <div id="box1"></div>
</body>
```

```
#box1 {
  height: 100px;
  width: 100px;
  background-color: red;

  top: 0;
  left: 0;
}
```

Heading 1

Lorem ipsum, or lipsum as it is sometimes known, is dummy text used in laying out print, graphic or web designs. The passage is attributed to an unknown typesetter in the 15th century who is thought to have scrambled parts of Cicero's De Finibus Bonorum et Malorum for use in a type specimen book.

Heading 2

Morbi tristique senectus et netus et malesuada fames ac turpis. Risus sed vulputate odio ut enim blandit volutpat maecenas. Habitasse platea dictumst quisque sagittis purus. Enim sit amet venenatis urna.



MSKU CENG

CENG-3507 Web Development & Programming

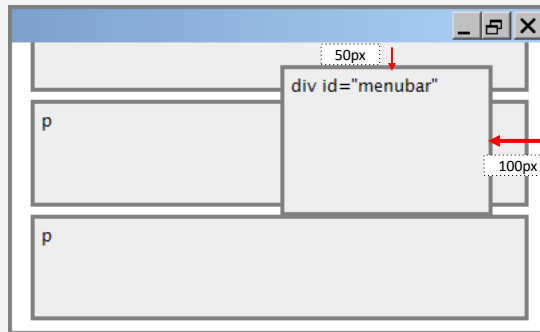
<https://codepen.io/pen?template=dyKYryM>

10

position: fixed

```
#menubar {
  position: fixed;
  top: 50px;
  right: 100px;
}
```

- For **fixed positioning**, the offset is the distance positioned relative to the viewport.
- The element does not move when scrolled.
- Element is **removed from normal document flow**, positioned on its own layer



Often used to implement UIs; control bars that shouldn't go away



MSKU CENG

CENG-3507 Web Development & Programming

11

position: fixed

<https://codepen.io/pen?template=VwdvReK>

```
#box1 {
  height: 50px;
  background-color:
    rgba(0, 0, 0, 0.5);

  position: fixed;
  top: 50px;
  left: 0;
  right: 0;
}
```

Heading 1

A puppy is a juvenile dog. Some puppies can weigh 1–3 lb (0.45–1.36 kg), while larger ones can weigh up to 15–23 lb (6.8–10.4 kg). All healthy puppies grow quickly after birth. A puppy's coat color may change as the puppy grows older, as is commonly seen in breeds such as the Yorkshire Terrier. In vernacular English, puppy refers specifically to dogs, while pup may often be used for other mammals such as seals, giraffes, guinea pigs, or even rats.

Heading 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Pharetra et ultrices neque ornare aenean euismod. Diam vulputate ut pharetra sit. Nunc mattis enim ut tellus elementum sagittis vitae. Eleifend mi in nulla posuere sollicitudin. Sagittis orci a scelerisque purus semper eget dui at tellus. Vitae suscipit tellus mauris a. Amet dictum sit amet justo. Diam maecenas ultrices mi eget mauris. Id eu nisl nunc mi. Adipiscing at in tellus integer feugiat scelerisque varius morbi enim. Non enim praesent elementum facilisis. Adipiscing elit dui tristique sollicitudin nibh sit. Nunc scelerisque viverra mauris in aliquam sem fringilla. Amet mattis vulputate enim nulla aliquet porttitor lacus luctus. Tincidunt tortor aliquam nulla facilisi cras fermentum odio. Facilisis magna etiam tempor orci eu. Et tortor at risus viverra adipiscing at. Eu ultrices vitae auctor eu augue ut lectus arcu bibendum. Convallis posuere morbi leo urna molestie at elementum.

Heading 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Pharetra et ultrices neque ornare aenean euismod. Diam vulputate ut pharetra sit. Nunc mattis enim ut tellus elementum sagittis vitae. Eleifend mi in nulla posuere sollicitudin. Sagittis orci a scelerisque purus semper eget dui at tellus. Vitae suscipit tellus mauris a. Amet dictum sit amet justo. Diam maecenas ultrices mi eget mauris. Id eu nisl



MSKU CENG

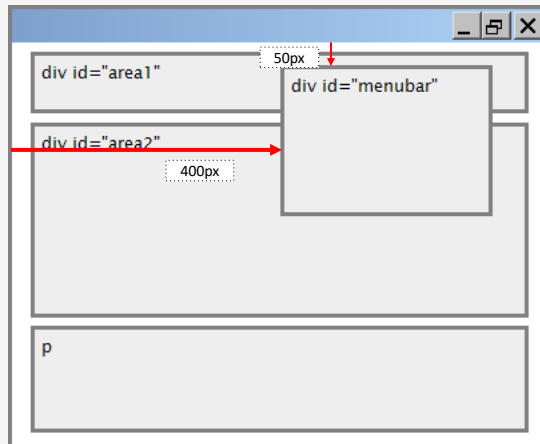
CENG-3507 Web Development & Programming

12

position: absolute

```
#menubar {
  position: absolute;
  left: 400px;
  top: 50px;
}
```

- For **absolute positioning**, the offset is the distance from the "containing element", which is the html element by default
- Element is removed from normal document flow, positioned on its own layer



MSKU CENG

CENG-3507 Web Development & Programming

13

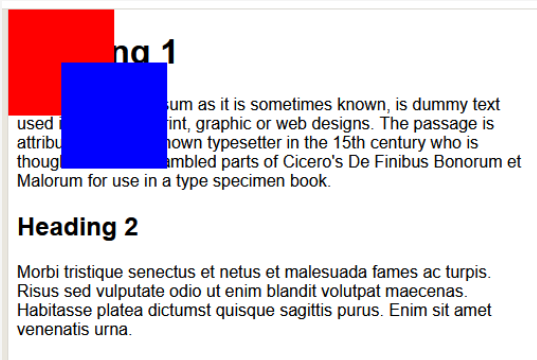
position: absolute

```
#box1 {
  height: 100px;
  width: 100px;
  background-color: red;

  position: absolute;
  top: 0;
  left: 0;
}

#box2 {
  height: 100px;
  width: 100px;
  background-color: blue;

  position: absolute;
  top: 50px;
  left: 50px;
}
```



MSKU CENG

CENG-3507 Web Development & Programming

<https://codepen.io/pen?template=mdKeoOw>

14

position: relative

For `position: relative`; the element is placed where it would normally be placed in the layout of the page, but shifted by the `top` / `left` / `bottom` / `right` values.

```
#box2 {
  height: 100px;
  width: 100px;
  background-color: blue;

  position: relative;
  top: 50px;
  left: 50px;
}
```

Heading 1

Lorem ipsum, or lipsum as it is sometimes known, is dummy text used in laying out print, graphic or web designs. The passage is attributed to an unknown typesetter in the 15th century who is thought to have scrambled parts of Cicero's De Finibus Bonorum et Malorum for use in a type specimen book.

Heading 2

Morbi tristique senectus et netus et malesuada fames ac turpis. Risus sed vulputate odio ut enim blandit volutpat maecenas. Habitasse platea dictumst quisque sagittis purus. Enim sit amet venenatis urna.



MSKU CENG

CENG-3507 Web Development & Programming

15

Relative absolute positioning

Let's revisit the definition of absolute positioning:

- **absolute**: a fixed position within its "containing element"
- The containing element is the viewport by default

You can change the containing element by setting "**position: relative**;" on some parent of your absolutely positioned element!



MSKU CENG

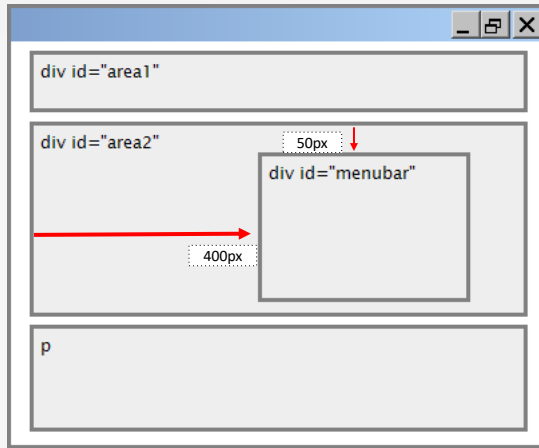
CENG-3507 Web Development & Programming

16

Relative absolute positioning

```
#area2 {
  position: relative;
}

#menubar {
  position: absolute;
  left: 400px;
  top: 50px;
}
```



Offsets are relative to the first parent element that has **position: relative** which in this case is **#area2**



MSKU CENG

CENG-3507 Web Development & Programming

17

Common use case: Overlay

```
<header>
  <div id="overlay"></div>
</header>

header {
  background-image: url(https://...);
  background-size: cover;
  height: 300px;
  position: relative;
}

#overlay {
  background-color:
    rgba(0, 0, 0, 0.3);
  position: absolute;
  top: 0;
  bottom: 0;
  height: 100%;
  width: 100%;
}
```



<https://codepen.io/pen?template=jOKbJwX>



MSKU CENG

CENG-3507 Web Development & Programming

18

Let's revisit Squarespace again!
([link to solution](https://codepen.io/pen?template=PoaPgbV))

<https://codepen.io/pen?template=PoaPgbV>



MSKU CENG

CENG-3507 Web Development & Programming

19

Random useful CSS



MSKU CENG

CENG-3507 Web Development & Programming

20

calc

You can use the [calc](#) CSS function to define numeric values in terms of expressions:

```
width: calc(50% - 10px);
```

```
width: calc(100% / 6);
```

([MDN details of calc](#))

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



MSKU CENG

CENG-3507 Web Development & Programming

21

CSS variables

Variables are a brand-new CSS feature
([caniuse](#)/ <http://caniuse.com/#search=css%20variables>)

```
:root {
  --primary-color: hotpink;
}
```

```
h1 {
  background-color: var(--primary-color);
}
```



MSKU CENG

CENG-3507 Web Development & Programming

22

background properties

An easy way to render images stretched and cropped to a given size: set it as a background image for an element.

`background-image: url(background.png);`

```
header {
  background-image: url(https://images.freecreatives.com/wp-content/uploads/2016/10/Free-HD-Stock-Image.jpg);
  background-size: cover;
  height: 300px;
  position: relative;
}
```



MSKU CENG

CENG-3507 Web Development & Programming

23

background properties

You can then use [additional background properties](#) to further style it:

```
background-size: cover;
background-size: contain;
background-repeat: no-repeat;
background-position: top;
background-position: center;
```

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Background_and_Borders



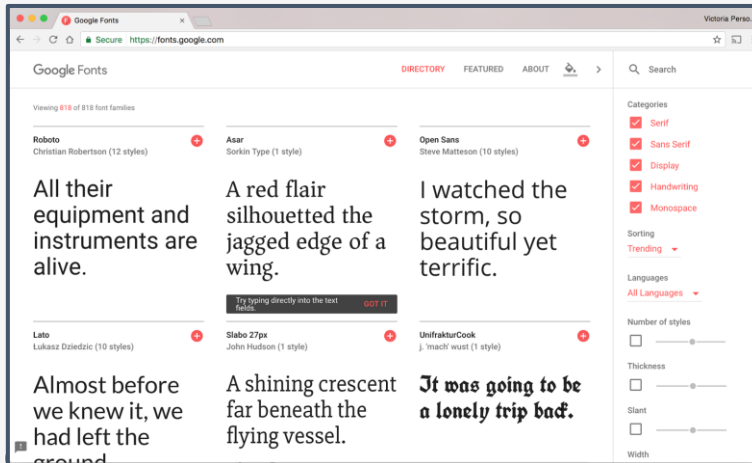
MSKU CENG

CENG-3507 Web Development & Programming

24

Web Fonts

You can use [Google Fonts](https://fonts.google.com) to choose from better fonts:



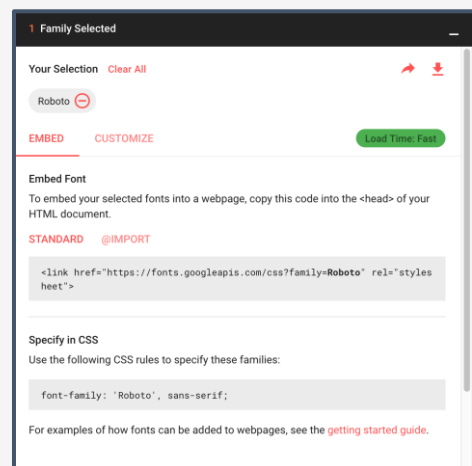
25

Web Fonts

You can use [Google Fonts](https://fonts.google.com) to choose from better fonts:

The instructions are pretty self-explanatory:

Basically, add the given `<link>` tag into the `<head>` section of your page alongside your other CSS files.



26

Aside: Fallback fonts

Notice that the Google Font example shows a comma-separated list of values for `font-family`:

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

- Each successive font listed is a fallback, i.e. the font that will be loaded if the previous font could not be loaded
- There are also six [generic font names](#), which allows the browser to choose the font based on intent + fonts available on the OS.
- It's good practice to list a generic font at the end of all your `font-family` declarations.


MSKU CENG

CENG-3507 Web Development & Programming

27

Hosted fonts with @font-face

You can also load your own font via [@font-face](#):

- Give it your own font name
- Link to where the font file is found

```
<body>
  <h1>Always and Forever</h1>
</body>
```

```
@font-face {
  font-family: "My Custom Font";
  src: url("https://s3-us-west-2.amazonaws.com/...");
}

body {
  font-family: "My Custom Font", serif;
}
```

Always and Forever


MSKU CENG

CENG-3507 Web Development & Programming

28

Mobile web



MSKU CENG

CENG-3507 Web Development & Programming

29

Q: What does it look like on a phone?

Heading 1

A puppy is a juvenile dog. Some puppies can weigh 1–3 lb (0.45–1.36 kg), while larger ones can weigh up to 15–23 lb (6.8–10.4 kg). All healthy puppies grow quickly after birth. A puppy's coat color may change as the puppy grows older, as is commonly seen in breeds such as the Yorkshire Terrier. In vernacular English, puppy refers specifically to dogs, while pup may often be used for other mammals such as seals, giraffes, guinea pigs, or even rats.

Heading 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Pharetra et ultrices neque ornare aenean euismod. Diam vulputate ut pharetra sit. Nunc mattis enim ut tellus elementum sagittis vitae. Eleifend mi in nulla posuere sollicitudin. Sagittis orci a scelerisque purus semper eget duis at tellus. Vitae suscipit tellus mauris a. Amet dictum sit amet justo. Diam maecenas ultricies mi eget mauris. Id eu nisl nunc mi. Adipiscing at in tellus integer feugiat scelerisque varius morbi enim. Non enim praesent elementum facilisis. Adipiscing elit duis tristique sollicitudin nibh sit. Nunc scelerisque viverra mauris in aliquam sem fringilla. Amet mattis vulputate enim nulla aliquet porttitor lacus luctus. Tincidunt tortor aliquam nulla facilisi cras fermentum odio. Facilisis magna etiam tempor orci eu. Et tortor at risus viverra adipiscing at. Eu ultrices vitae auctor eu augue ut lectus arcu bibendum. Convallis posuere morbi leo urna molestie at elementum.

Heading 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Pharetra et ultrices neque ornare aenean euismod. Diam vulputate ut pharetra sit. Nunc mattis enim ut tellus elementum sagittis vitae. Eleifend mi in nulla posuere sollicitudin. Sagittis orci a scelerisque purus semper eget duis at tellus. Vitae suscipit tellus mauris a. Amet dictum sit amet justo. Diam maecenas ultricies mi eget mauris. Id eu nisl nunc mi. Adipiscing at in tellus integer feugiat scelerisque varius morbi enim. Non enim praesent elementum facilisis. Adipiscing elit duis tristique sollicitudin nibh sit. Nunc scelerisque viverra mauris in aliquam sem fringilla. Amet mattis vulputate enim nulla aliquet porttitor lacus luctus. Tincidunt tortor aliquam nulla facilisi cras fermentum odio. Facilisis magna etiam tempor orci eu. Et tortor at risus viverra adipiscing at. Eu ultrices vitae auctor eu augue ut lectus arcu bibendum. Convallis posuere morbi leo urna molestie at elementum.



MSKU CENG

CENG-3507 Web Development & Programming

30

Terribly small and hard to read.



MSKU CENG

CENG-3507 Web Development & Programming

31

Responsive web design

We want to write our CSS in a way that can look nice in a wide range of screen sizes:

- 27" desktop monitor
- Macbook Air
- Samsung Galaxy S7
- iPhone 7
- iPad

Q: How do we do this?

Do we need to write totally different CSS for every screen size?!



MSKU CENG

CENG-3507 Web Development & Programming

32

Mobile sizing



Unless directed otherwise via HTML or CSS cues, mobile browsers render web pages at a **desktop screen width** (~1000px), then "zooms out" until the entire page fits on screen.

(That's why you sometimes get web pages with teeny-tiny font on your phone: these webpages have not added support for mobile.)

([Read more on how this works](#))



MSKU CENG

CENG-3507 Web Development & Programming

33

Meta viewport tag

To prevent phone browsers from rendering the page at desktop width and zooming out, use the **meta viewport tag**:

```
<meta name="viewport"
content="width=device-width, initial-scale=1">
```

This belongs in the `<head>` section of your HTML.
(Same section as the `<title>`, `<link>`, and other metadata elements.)



MSKU CENG

CENG-3507 Web Development & Programming

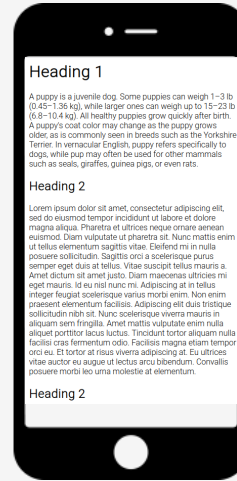
34

Meta viewport tag

Without the meta viewport tag



With the meta viewport tag



MSKU CENG

CENG-3507 Web Development & Programming

35

Meta viewport tag

```
<meta name="viewport"
content="width=device-width, initial-scale=1">
```

- **name=viewport**: "Browser, I am going to tell you how I want the viewport to look."
- **width=device-width**: "The viewport's width should always start at the device's width." (i.e., don't do that thing on mobile where you render the page at the desktop's width)
- **initial-scale=1**: "Start at zoom level of 100%."



MSKU CENG

CENG-3507 Web Development & Programming

36

Meta viewport tag

```
<meta name="viewport"
content="width=device-width, initial-scale=1">
```

(You should pretty much always include this tag in your HTML.)



MSKU CENG

CENG-3507 Web Development & Programming

37

Making adjustments

The meta viewport tag gets us almost all the way there, but we want to make a few adjustments.

For example, the margin seems a tad too big on mobile. Can we set a different margin property for mobile?

Heading 1

A puppy is a juvenile dog. Some puppies can weigh 1–3 lb (0.45–1.36 kg), while larger ones can weigh up to 15–23 lb (6.8–10.4 kg). All healthy puppies grow quickly after birth. A puppy's coat color may change as the puppy grows older, as is commonly seen in breeds such as the Yorkshire Terrier. In vernacular English, puppy refers specifically to dogs, while pup may often be used for other mammals such as seals, giraffes, guinea pigs, or even rats.

Heading 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Pharetra et ultrices neque ornare aenean euismod. Diam vulputate ut pharetra sit. Nunc mattis enim ut tellus elementum sagittis vitae. Eleifend mi in nulla posuere sollicitudin. Sagittis orci a scelerisque purus semper eget duis at tellus. Vitae suscipit tellus mauris a. Amet dictum sit amet justo. Diam maecenas ultricies mi eget mauris. Id eu nisl nunc mi. Adipiscing at in tellus integer feugiat scelerisque varius morbi enim. Non enim praesent elementum facilisis. Adipiscing elit duis tristique sollicitudin nibh sit. Nunc scelerisque viverra mauris in aliquam sem fringilla. Amet mattis vulputate enim nulla aliquet porttitor lacus luctus. Tincidunt tortor aliquam nulla facilisi cras fermentum odio. Facilisis magna etiam tempor orci eu. Et tortor at risus viverra adipiscing at. Eu ultrices vitae auctor eu augue ut lectus arcu bibendum. Convallis posuere morbi leo urna molestie at elementum.

Heading 2



MSKU CENG

CENG-3507 Web Development & Programming

38

CSS media queries

You can define a **CSS media query** in order to change style rules based on the characteristics of the device:

```
@media (max-width: 500px) {  
  body {  
    margin: 0 2px;  
  }  
}
```

You can create [much more complex](https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries) media queries as well.

https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries

**MSKU CENG**

CENG-3507 Web Development & Programming

39

Development strategies

Practical question: **How do you test mobile layouts?**

- Do you upload your HTML+CSS somewhere online and navigate to that URL on your phone?
- Is there a way to connect your phone to your local device?
- Do you run it in an Android/iOS emulator?
- Other?!

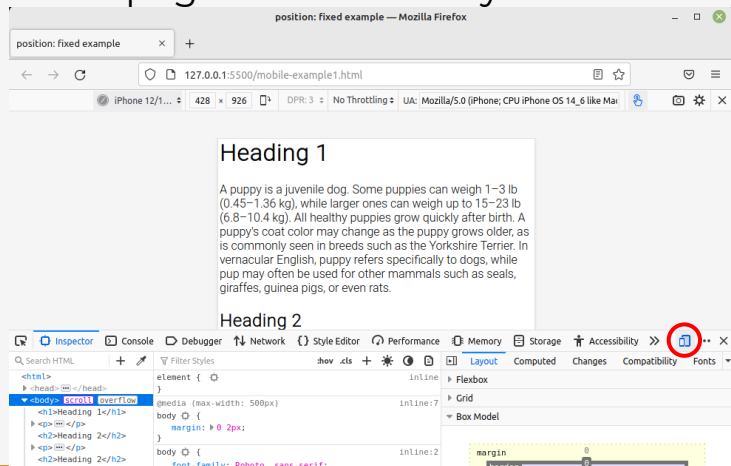
**MSKU CENG**

CENG-3507 Web Development & Programming

40

Browser's device mode in Web Developer Tools

You can simulate a web page in a mobile layout via **device mode**:



MSKU CENG

CENG-3507 Web Development & Programming

41

Chrome device mode

Advantages of device mode:

- Super convenient
- Mostly accurate

Disadvantages of device mode:

- Not always accurate - iPhone particularly an issue
- A little buggy
- Doesn't simulate performance issues

You should always test on real devices, too.



MSKU CENG

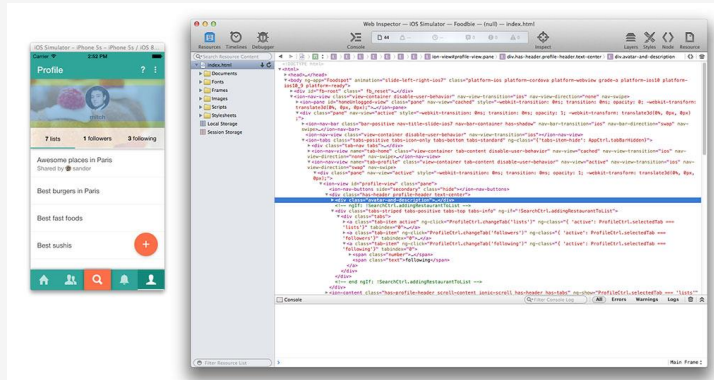
CENG-3507 Web Development & Programming

42

Safari remote debugging

<http://developer.telerik.com/featured/a-concise-guide-to-remote-debugging-on-ios-android-and-windows-phone/>

If you have an iPhone, you can debug web pages on your phone via Safari remote debugging.



MSKU CENG

CENG-3507 Web Development & Programming

43

Relative font sizes: percent, em, rem



MSKU CENG

CENG-3507 Web Development & Programming

44

Relative units

Whenever possible, it's best to use **relative units** (like percentage) instead of absolute units (like px).

Advantages:

- More likely to work on different screen sizes
- Easier to reason about; fewer magic numbers
10% / 80% / 10% vs 122px / 926px / 122px

Q: Should we be using relative units on font-size?



MSKU CENG

CENG-3507 Web Development & Programming

45

Relative font sizes: percent

You can define font sizes in terms of percentage:

```
<body>
  <h1>This is 60px</h1>
  <p>This is 15px</p>
</body>
```

```
body {
  font-size: 30px;
}

h1 {
  font-size: 200%;
}

p {
  font-size: 50%;
}
```

This is 60px

This is 15px



MSKU CENG

CENG-3507 Web Development & Programming

46

Relative font sizes: percent

Percent on font-size behaves exactly like percentage on width and height, in that it's relative to the parent:

```
<div>
  This is 60px
  <p>This is 45px</p>
</div>
```

```
body {
  font-size: 30px;
}

div {
  font-size: 200%;
}

p {
  font-size: 75%;
}
```

This is 60px

This is 45px



MSKU CENG

CENG-3507 Web Development & Programming

47

Relative font sizes: percent

Percent on font-size behaves exactly like percentage on width and height, in that it's relative to the parent:

```
<div>
  This is 60px
  <p>This is 45px</p>
</div>
```

```
body {
  font-size: 30px;
}

div {
  font-size: 200%;
}

p {
  font-size: 75%;
}
```

p is 75% of its parent, which is 200% of 30px.

p's size: $.75 * 2 * 30 = 45\text{px}$

This is 60px

This is 45px



MSKU CENG

CENG-3507 Web Development & Programming

48

Relative font sizes: em

But instead of percentages, relative font sizes are usually defined in terms of **em**:

- **em** represents the calculated font-size of the element
 - **1em** = the inherited font size
 - **2em** = 2 times the inherited font size

In other words,
font-size: 1em; is the same as font-size: 100%;



MSKU CENG

CENG-3507 Web Development & Programming

49

Relative font sizes: em

```
<body>
  <h1>This is 60px</h1>
  <p>This is 15px</p>
</body>
```

```
body {
  font-size: 30px;
}

div {
  font-size: 2em;
}

p {
  font-size: .5em;
}
```

This is 60px

This is 15px



MSKU CENG

CENG-3507 Web Development & Programming

50

Relative font sizes: em

```
<div>
  This is 60px
  <p>This is 45px</p>
</div>
```

```
body {
  font-size: 30px;
}

div {
  font-size: 2em;
}

p {
  font-size: .75em;
}
```

This is 60px

This is 45px



MSKU CENG

CENG-3507 Web Development & Programming

51

Relative font sizes: em

```
<div>
  This is 60px
  <p>This is 45px</p>
</div>
```

```
body {
  font-size: 30px;
}

div {
  font-size: 2em;
}

p {
  font-size: .75em;
}
```

p's inherited font size is
2em, which is 60px. So
 0.75em is $0.75 \times 60 = 45\text{px}$.

This is 60px

This is 45px



MSKU CENG

CENG-3507 Web Development & Programming

52

```
<body>
  This is
  <h1>
    <strong>120px</strong>
  </h1>
</body>
```

```
body {
  font-size: 30px;
}

strong {
  font-size: 2em;
}
```

This is

120px

Why is 120px and not 60px?



MSKU CENG

CENG-3507 Web Development & Programming

53

```
<body>
  This is
  <h1>
    <strong>120px</strong>
  </h1>
</body>
```

```
body {
  font-size: 30px;
}

strong {
  font-size: 2em;
}
```

This is

120px

```
h1 {                                user agent stylesheet
  display: block;
  font-size: 2em;
  -webkit-margin-before: 0.67em;
  -webkit-margin-after: 0.67em;
  -webkit-margin-start: 0px;
  -webkit-margin-end: 0px;
  font-weight: bold;
}
```

The Chrome Inspector show that the default browser font-size for h1 is 2em. So it's $30 * 2 * 2 = 120px$.



MSKU CENG

CENG-3507 Web Development & Programming

54

Relative font sizes: rem

If you **do not** want your relative font sizes to compound through inheritance, use `rem`:

- `rem` represents the font-size of the root element
 - `1rem` = the root (`html` tag) font size
 - `2rem` = 2 times root font size



MSKU CENG

CENG-3507 Web Development & Programming

55

Relative font sizes: rem

```
<body>
  <div>
    This is 60px
    <p>This is 22.5px</p>
  </div>
</body>
```

```
html {
  font-size: 30px;
}

div {
  font-size: 2rem;
}

p {
  font-size: .75rem;
}
```

This is 60px

This is 22.5px



MSKU CENG

CENG-3507 Web Development & Programming

56

Relative font sizes: rem

```
<body>
  <div>
    This is 60px
    <p>This is 22.5px</p>
  </div>
</body>
```

```
html {
  font-size: 30px;
}

div {
  font-size: 2rem;
}

p {
  font-size: .75rem;
}
```

font-size is set on the html element, not body (or any other tag)

This is 60px

This is 22.5px



MSKU CENG

CENG-3507 Web Development & Programming

57

Relative font sizes: rem

```
<body>
  <div>
    This is 60px
    <p>This is 22.5px</p>
  </div>
</body>
```

```
html {
  font-size: 30px;
}

div {
  font-size: 2rem;
}

p {
  font-size: .75rem;
}
```

.75em is calculated from the root, which is 30px, so $30 * .75 = 22.5$ px.

This is 60px

This is 22.5px



MSKU CENG

CENG-3507 Web Development & Programming

58

Relative font conclusions

Use relative fonts for the same purpose as using relative heights and widths:

- Prefer **em** and **rem** over percentages
 - Not for any deep reason, but **em** is meant for font so it's semantically cleaner
- Use **rem** to avoid compounding sizes
- In addition to **font-size**, consider **em/rem** for:
 - **line-height**
 - **margin-top**
 - **margin-bottom**



MSKU CENG

CENG-3507 Web Development & Programming

59

Mobile summary

- Always add the **meta viewport tag**
- Use **@media queries** to add styles for devices with certain characteristics, such as screen width
- Use the **Chrome Device Mode** to simulate mobile rendering on desktop
- For height and width, prefer percentages
- For fonts, prefer **em** and **rem**
- Try to **minimize dependent rules** (Changing the width of one container force you to change 15 other properties to look right)

More on [responsive web design](#)



MSKU CENG

CENG-3507 Web Development & Programming

60

How do web pages work again?

**MSKU CENG**

CENG-3507 Web Development & Programming

61

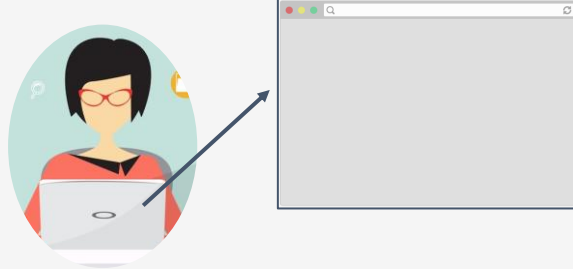
You are on
your laptop

**MSKU CENG**

CENG-3507 Web Development & Programming

62

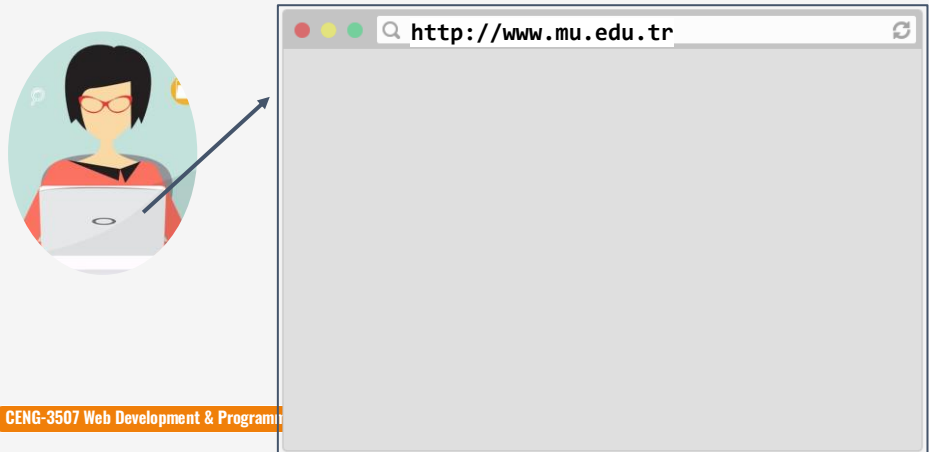
Your laptop is running
a web browser, e.g.
Chrome

**MSKU CENG**

CENG-3507 Web Development & Programming

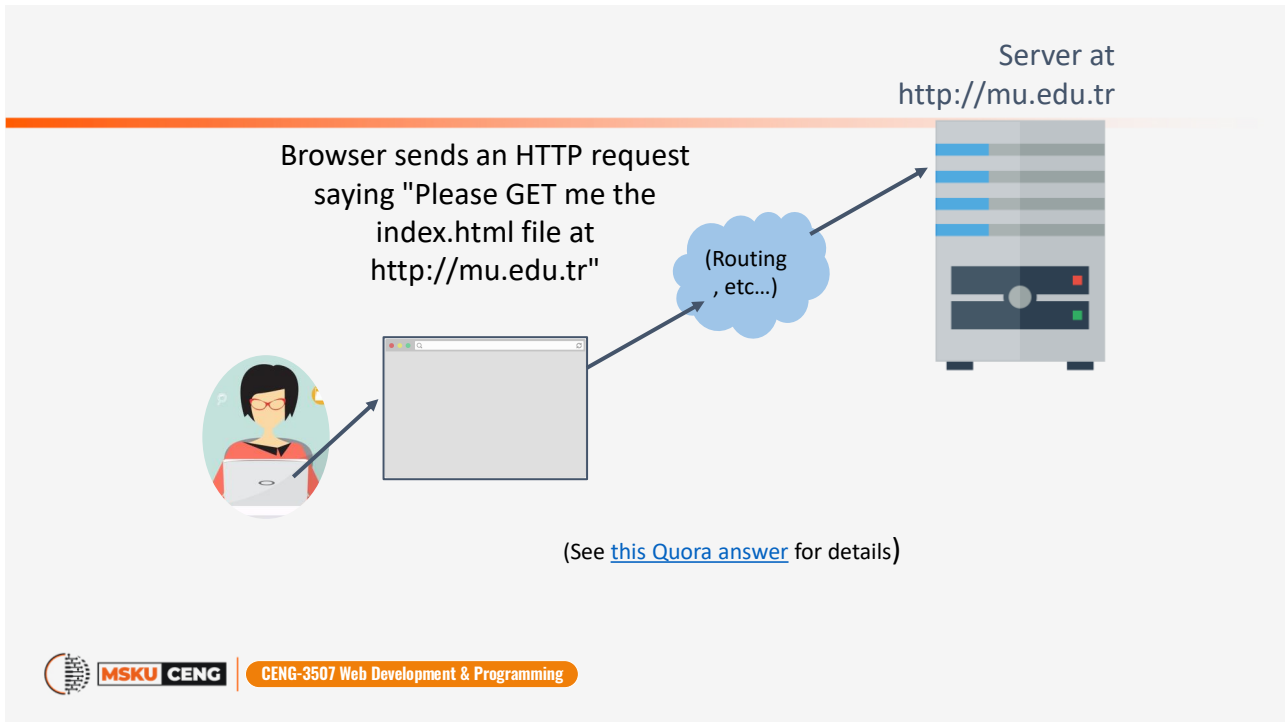
63

You type a URL in the address bar
and hit "enter"

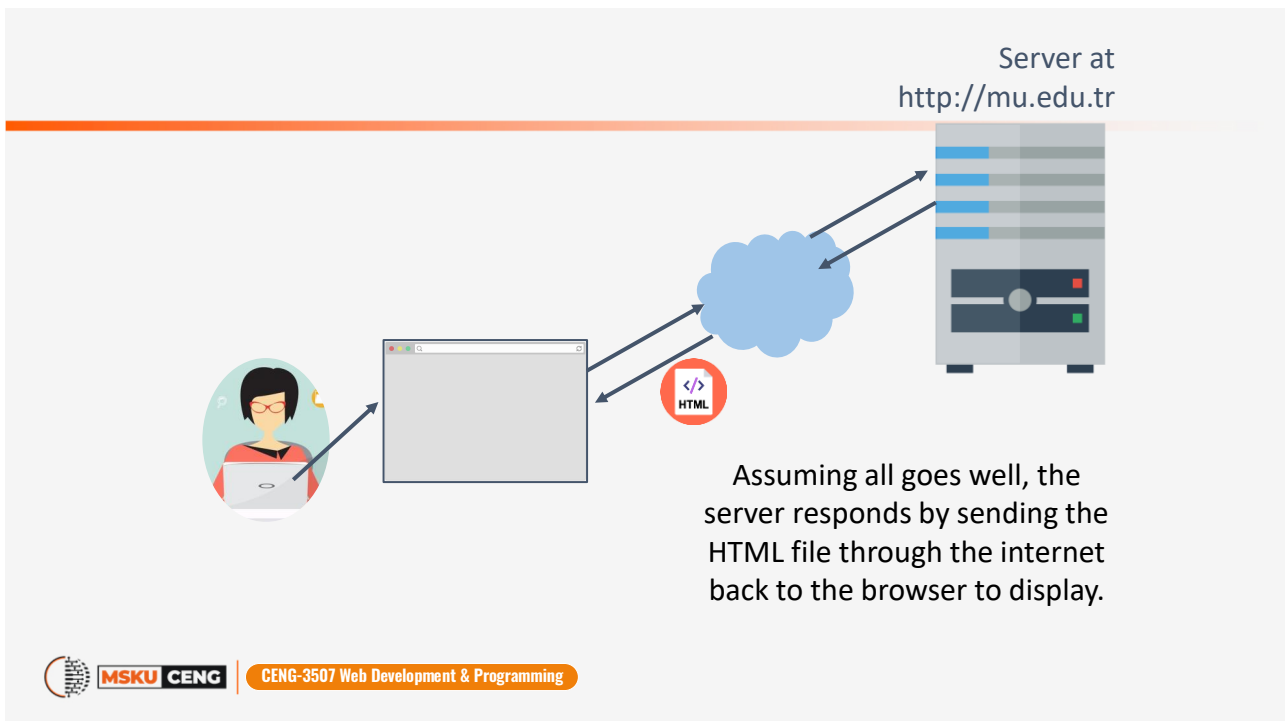
**MSKU CENG**

CENG-3507 Web Development & Programming

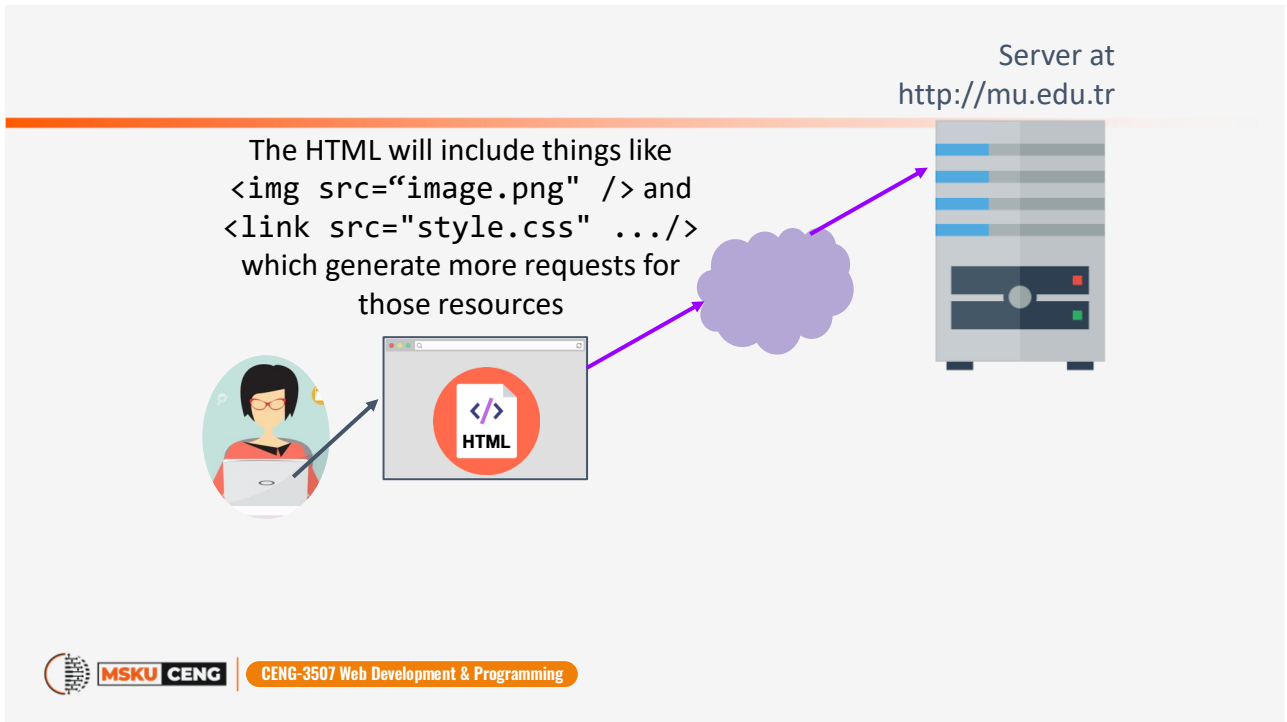
64



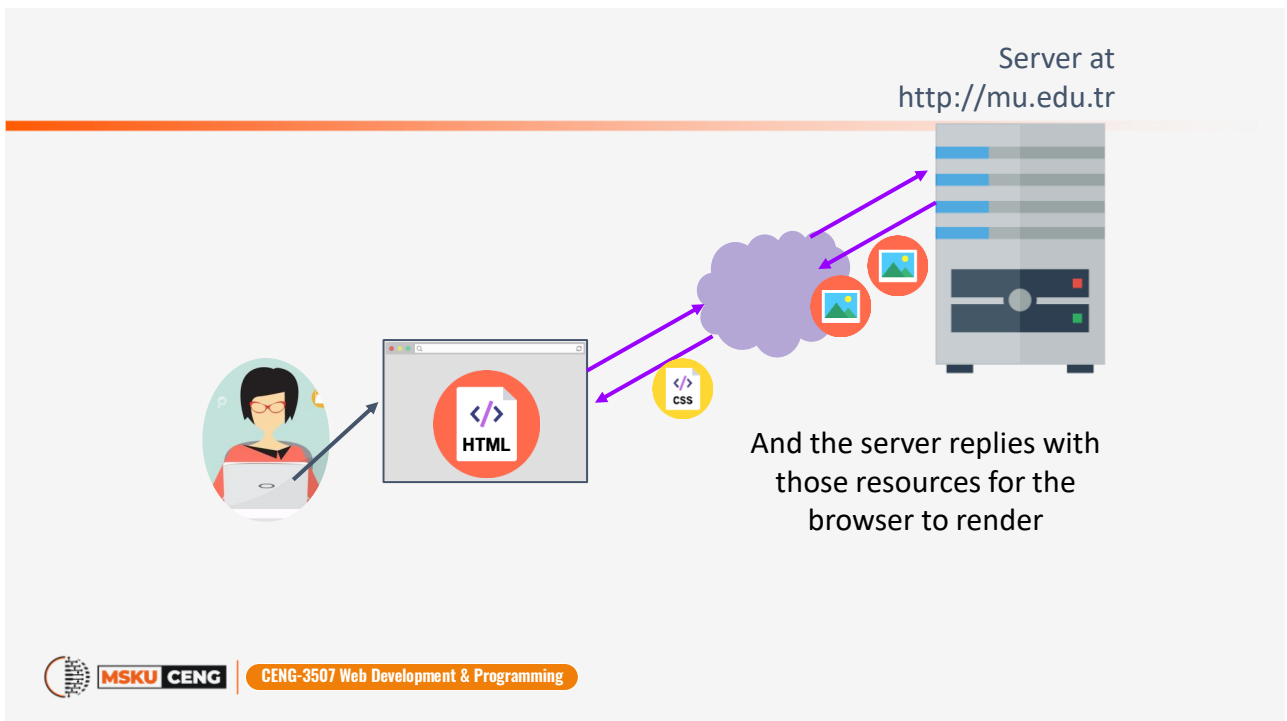
65



66

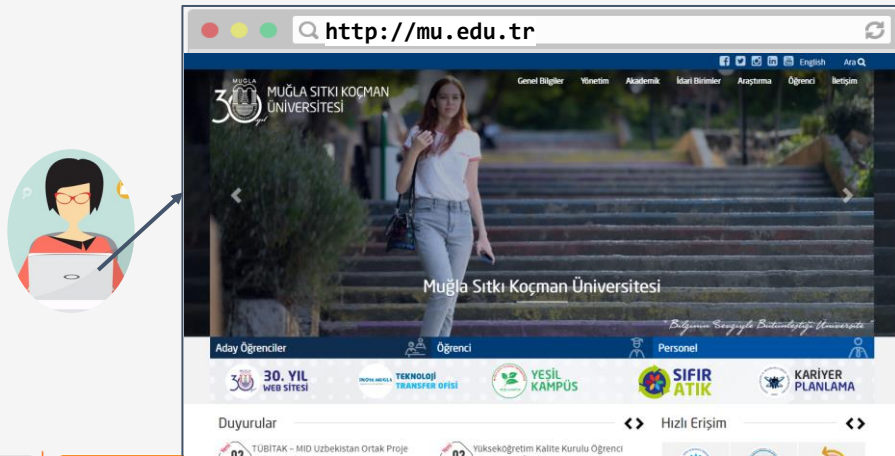


67



68

Finally, when all resources are loaded, we see the loaded web page



69



Describes the
content and
structure of
the page

+



Describes the
appearance
and style of
the page

produces



A web page...
that doesn't do
anything



MSKU CENG

CENG-3507 Web Development & Programming

70

What we've learned so far

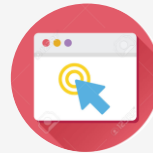
We've learned how to build web pages that:

- Look the way we want them to
- Can link to other web pages
- Display differently on different screen sizes



But we don't know how build web pages that **do** anything:

- Get user input
- Save user input
- Show and hide elements when the user interacts with the page
- etc.



MSKU CENG

CENG-3507 Web Development & Programming

71

JavaScript



MSKU CENG

CENG-3507 Web Development & Programming

72

JavaScript

JavaScript is a programming language.

It is currently the only programming language that your browser can execute natively. (There are [efforts](#) to change that.)

Therefore if you want to make your web pages do stuff, you must use JavaScript: There are no other options.



MSKU CENG

CENG-3507 Web Development & Programming

73

JavaScript

- Created in 1995 by Brendan Eich
(co-founder of Mozilla;)
- JavaScript has nothing to do with Java
 - Literally named that way for
- The first version was written in 10 days
- Several fundamental language decisions were made because of company politics and not technical reasons



MSKU CENG

CENG-3507 Web Development & Programming

74

JavaScript

In other words:

**JavaScript is messy and full of drama...
and our only option.**

(though it's gotten much, much better in the last few years)



MSKU CENG

CENG-3507 Web Development & Programming

75

JavaScript

This week:

- Simple, old-school JavaScript
- Mostly **not** best practice
 - Everything in global scope
 - No classes / modules
 - Will result in a big mess if you code this way for anything but very small projects
- (But easy to get started)

Next week: Modern JavaScript

- More disciplined and based on best practices
- Even more "opinionated"



MSKU CENG

CENG-3507 Web Development & Programming

76

JavaScript in the browser



MSKU CENG

CENG-3507 Web Development & Programming

77

Code in web pages

HTML can embed JavaScript files into the web page via the `<script>` tag.

```
<!DOCTYPE html>
<html>
  <head>
    <title>CS 193X</title>
    <link rel="stylesheet" href="style.css" />
    <script src="filename.js"></script>
  </head>
  <body>
    ... contents of the page...
  </body>
</html>
```



MSKU CENG

CENG-3507 Web Development & Programming

78

console.log

You can print log messages in JavaScript by calling `console.log()`:

script.js

```
console.log('Hello, world!');
```

This JavaScript's equivalent of Java's `System.out.println`, `print`, `printf`, etc.



MSKU CENG

CENG-3507 Web Development & Programming

79

How does JavaScript get loaded?



MSKU CENG

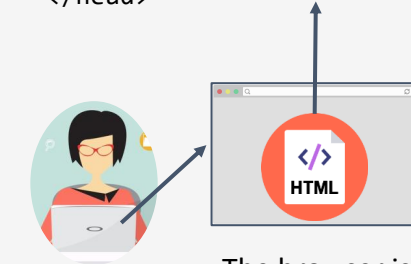
CENG-3507 Web Development & Programming

80


```

<head>
  <title>CS 193X</title>
  <link rel="stylesheet" href="style.css" />
  ➔ <script src="script.js"></script>
</head>

```



The browser is parsing the HTML file, and gets to a script tag, so it knows it needs to get the script file as well.



MSKU CENG

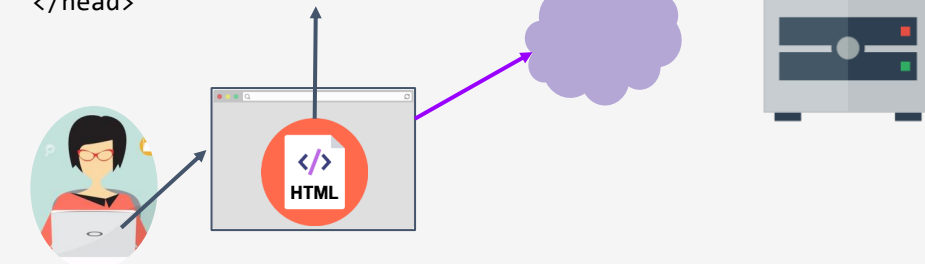
CENG-3507 Web Development & Programming

81

```

<head>
  <title>CS 193X</title>
  <link rel="stylesheet" href="style.css" />
  ➔ <script src="script.js"></script>
</head>

```



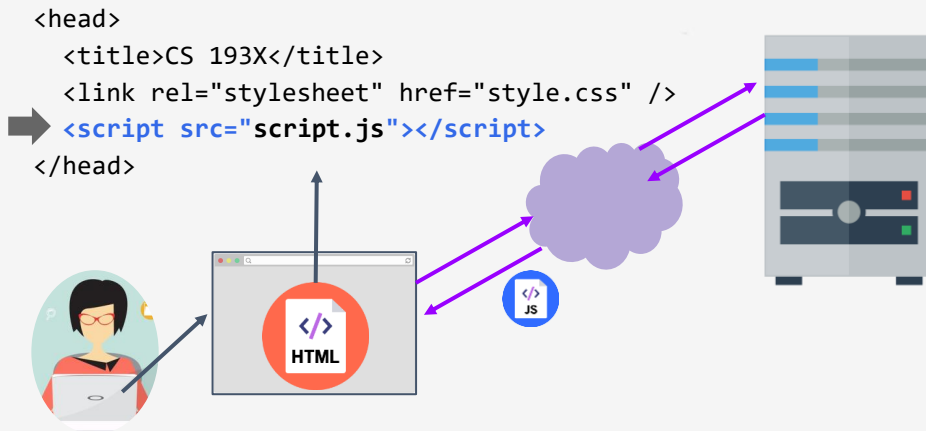
The browser makes a request to the server for the script.js file, just like it would for a CSS file or an image...



MSKU CENG

CENG-3507 Web Development & Programming

82



And the server responds with the JavaScript file, just like it would with a CSS file or an image...



MSKU CENG

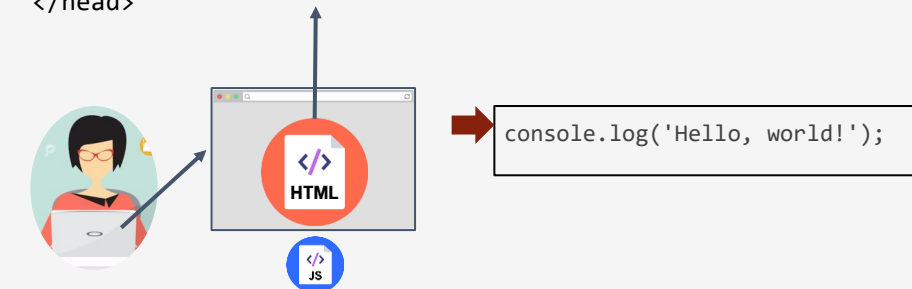
CENG-3507 Web Development & Programming

83

```

<head>
  <title>CS 193X</title>
  <link rel="stylesheet" href="style.css" />
  <script src="script.js"></script>
</head>

```



Now at this point, the JavaScript file will execute "**client-side**", or in the browser on the user's computer.



MSKU CENG

CENG-3507 Web Development & Programming

84

JavaScript execution

There is **no "main method"**

- The script file is executed from top to bottom.

There's **no compilation** by the developer

- JavaScript is compiled and executed on the fly by the browser

(Note that this is slightly different than being "interpreted": see [just-in-time \(JIT\) compilation](#))



MSKU CENG

CENG-3507 Web Development & Programming

85

first-js.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>First JS Example</title>
    <script src="script.js"></script>
  </head>
  <body>
  </body>
</html>
```



script.js

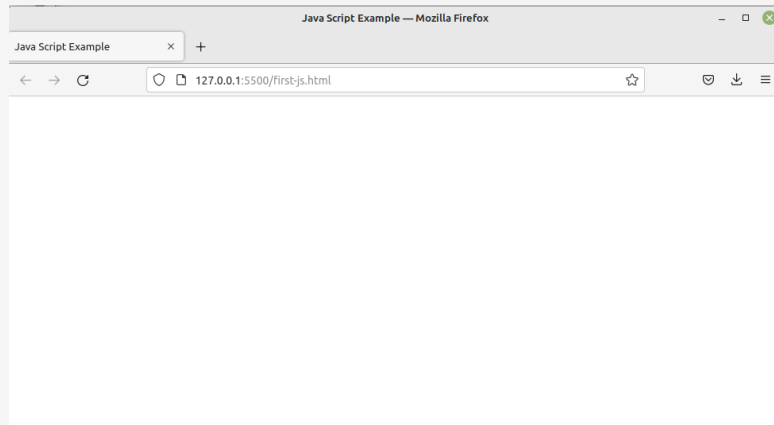
```
console.log('Hello, world!');
```



MSKU CENG

CENG-3507 Web Development & Programming

86



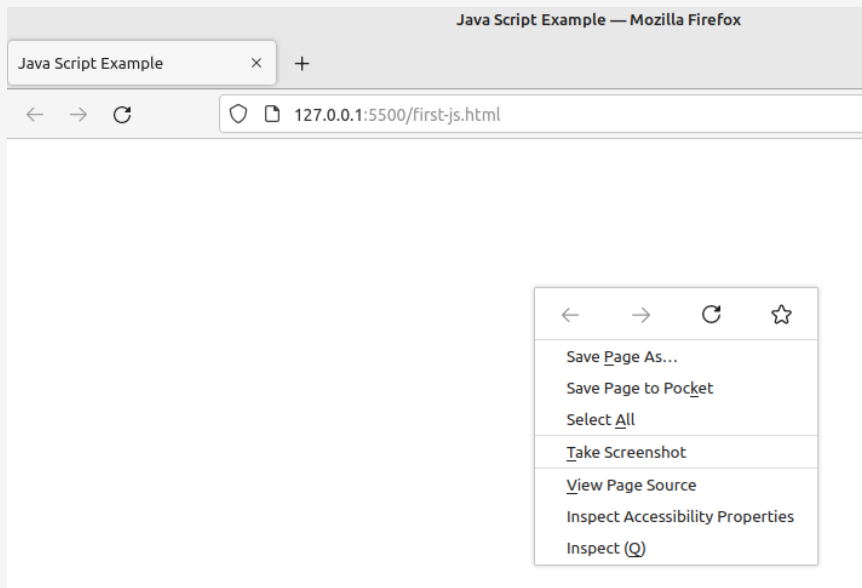
Nothing happened!



MSKU CENG

CENG-3507 Web Development & Programming

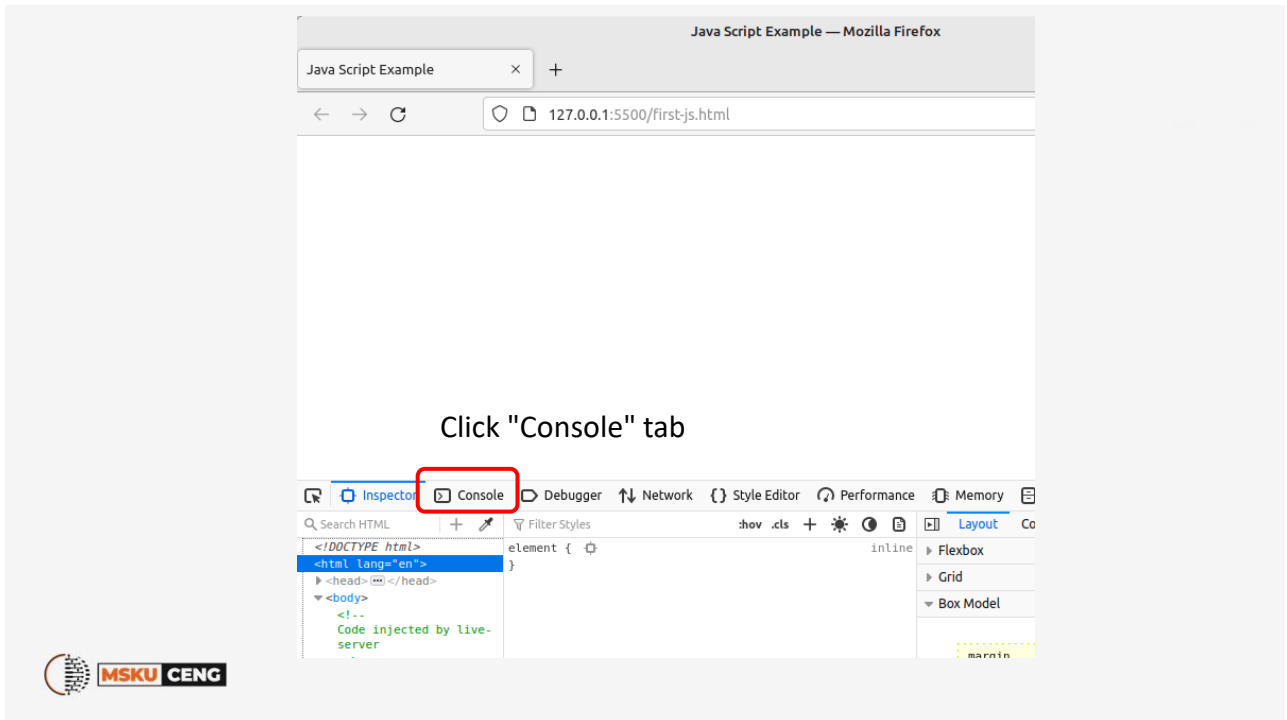
87



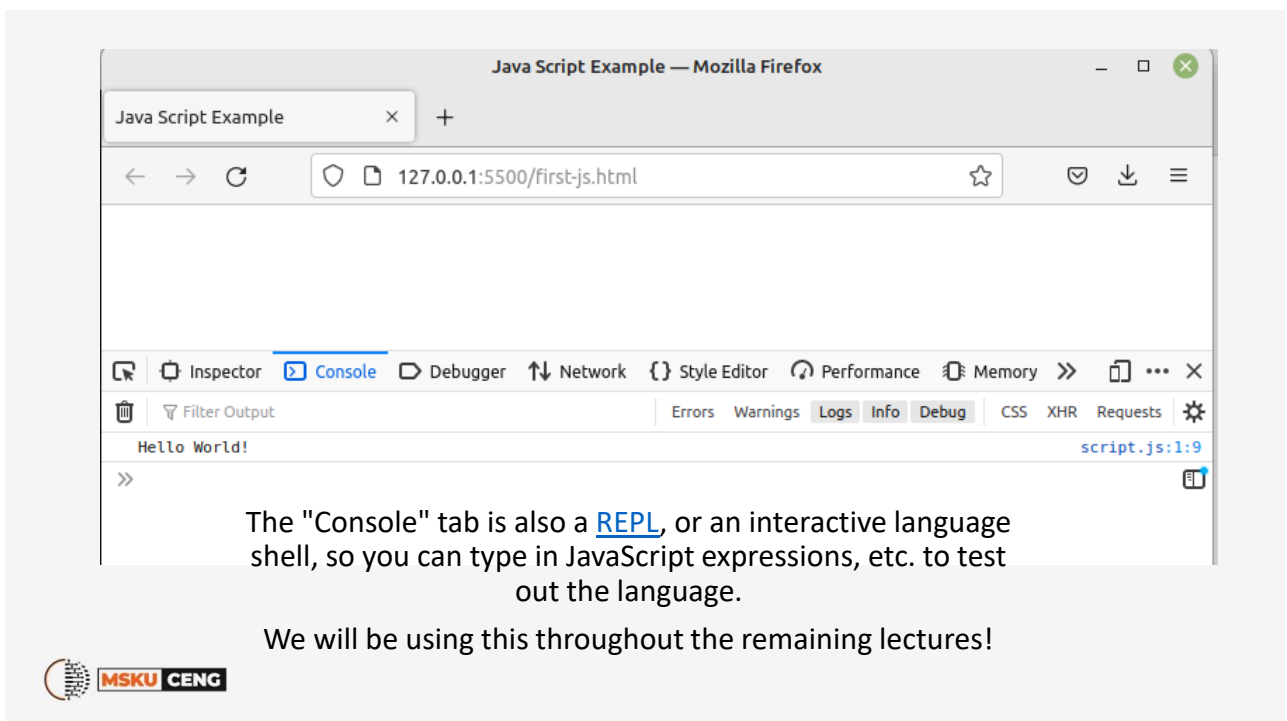
MSKU CENG

Right-click (or control-click on Mac) and choose "Inspect"

88



89



90

JavaScript language features



MSKU CENG

CENG-3507 Web Development & Programming

91

Same as Java/C++/C-style langs

for-loops:

```
for (let i = 0; i < 5; i++) { ... }
```

while-loops:

```
while (notFinished) { ... }
```

comments:

```
// comment or /* comment */
```

conditionals (if statements):

```
if (...) {  
    ...  
} else {  
    ...  
}
```



MSKU CENG

CENG-3507 Web Development & Programming

92

Functions

One way of defining a JavaScript function is with the following syntax:

```
function name() {
  statement;
  statement;
  ...
}
```



MSKU CENG

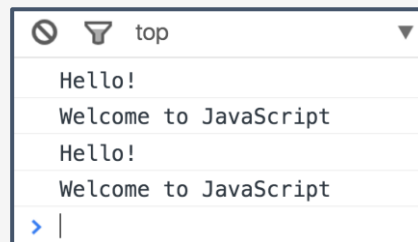
CENG-3507 Web Development & Programming

93

script.js

```
function hello() {
  console.log('Hello!');
  console.log('Welcome to JavaScript');
}

hello();
hello();
```



Console output



MSKU CENG

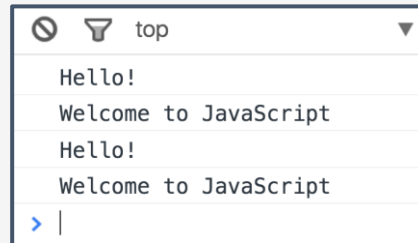
94

script.js

```
function hello() {
  console.log('Hello!');
  console.log('Welcome to JavaScript');
}

hello();
hello();
```

The browser "executes" the function definition first, but that just creates the hello function (and it doesn't run the hello function), similar to a variable declaration.



Console output



95

Q: Does this work?

script.js

```
hello();
hello();

function hello() {
  console.log('Hello!');
  console.log('Welcome to JavaScript');
}
```



CENG-3507 Web Development & Programming

96

script.js

<https://developer.mozilla.org/en-US/docs/Glossary/Hoisting>

```

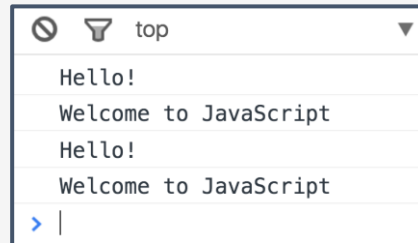
hello();
hello();

function hello() {
  console.log('Hello!');
  console.log('Welcome to JavaScript');
}

```

A: Yes, for this particular syntax.

This works because function declarations are "**hoisted**" ([mdn](#)). You can think of it as if the definition gets moved to the top of the scope in which it's defined (though that's not what actually happens).

**Console output**

97

script.js

```

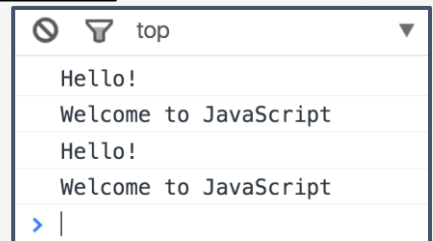
hello();
hello();

function hello() {
  console.log('Hello!');
  console.log('Welcome to JavaScript');
}

```

Console output**Caveats:**

- There are other ways to define functions that do not get hoisted
- Try not to rely on hoisting when coding. [It gets bad](#).

<http://www.adequatelygood.com/JavaScript-Scoping-and-Hoisting.html>

98

Variables: var, let, const

Declare a variable in JS with one of three keywords:

```
// Function scope variable
var x = 15;
// Block scope variable
let fruit = 'banana';
// Block scope constant; cannot be reassigned
const isHungry = true;
```

You do not declare the datatype of the variable before using it (["dynamically typed"](#))



MSKU CENG

CENG-3507 Web Development & Programming

99

Function parameters

```
function printMessage(message, times) {
  for (var i = 0; i < times; i++) {
    console.log(message);
  }
}
```

Function parameters are **not** declared with var, let, or const



MSKU CENG

CENG-3507 Web Development & Programming

100

Understanding var

```
function printMessage(message, times) {
  for (var i = 0; i < times; i++) {
    console.log(message);
  }
  console.log('Value of i is ' + i);
}
printMessage('hello', 3);
```

Q: What happens if we try to print "i" at the end of the loop?



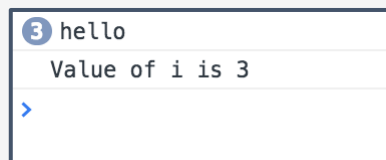
MSKU CENG

CENG-3507 Web Development & Programming

101

Understanding var

```
function printMessage(message, times) {
  for (var i = 0; i < times; i++) {
    console.log(message);
  }
  console.log('Value of i is ' + i);
}
printMessage('hello', 3);
```



The value of "i" is readable outside of the for-loop because variables declared with var have **function scope**.



MSKU CENG

102

Function scope with var

```
var x = 10;
if (x > 0) {
  var y = 10;
}
console.log('Value of y is ' + y);
```

Value of y is 10

- Variables declared with "var" have function-level scope and do not go out of scope at the end of blocks; only at the end of functions
- Therefore you can refer to the same variable after the block has ended (e.g. after the loop or if-statement in which they are declared)



MSKU CENG

CENG-3507 Web Development & Programming

103

Function scope with var

But you can't refer to a variable outside of the function in which it's declared.

```
function meaningless() {
  var x = 10;
  if (x > 0) {
    var y = 10;
  }
  console.log('y is ' + y);
}
meaningless();
console.log('y is ' + y); // error! ❌
```

y is 10

❌ ▶ Uncaught ReferenceError: y is not defined
at script.js:9



MSKU CENG

104

Understanding `let`

```
function printMessage(message, times) {
  for (let i = 0; i < times; i++) {
    console.log(message);
  }
  console.log('Value of i is ' + i);
}
printMessage('hello', 3);
```

Q: What happens if we try to print "i" at the end of the loop?



MSKU CENG

CENG-3507 Web Development & Programming

105

Understanding `let`

```
function printMessage(message, times) {
  for (let i = 0; i < times; i++) {
    console.log(message);
  }
  console.log('Value of i is ' + i);
}
printMessage('hello', 3);
```

`let` has block-scope so this results in an error

3 hello

✖ ▶ Uncaught ReferenceError: i is not defined
at printMessage (script.js:5)
at script.js:8

>



MSKU CENG

CENG-3507 Web Development & Programming

106

Understanding `const`

```
let x = 10;
if (x > 0) {
  const y = 10;
}
console.log(y); // error!
```

Like `let`, `const` also has block-scope, so accessing the variable outside the block results in an error

```
✖ ▶ Uncaught ReferenceError: y is not defined
    at script.js:5
```



MSKU CENG

CENG-3507 Web Development & Programming

107

Understanding `const`

```
const y = 10;
y = 0;           // error!
y++;            // error!
const list = [1, 2, 3];
list.push(4);    // OK
```

`const` declared variables cannot be reassigned.

However, it doesn't provide true `const` correctness, so you can still modify the underlying object

- (In other words, it behaves like Java's `final` keyword and not C++'s `const` keyword)



MSKU CENG

CENG-3507 Web Development & Programming

108

Contrasting with `let`

```
let y = 10;
y = 0;           // OK
y++;             // OK
let list = [1, 2, 3];
list.push(4);    // OK
```

`let` can be reassigned, which is the difference between `const` and `let`



MSKU CENG

CENG-3507 Web Development & Programming

109

Variables best practices

- Use `const` whenever possible.
- If you need a variable to be reassignable, use `let`.
- **Don't use `var`.**
 - You will see a ton of example code on the internet with `var` since `const` and `let` are relatively new.
 - However, `const` and `let` are [well-supported](#), so there's no reason not to use them.

(This is also what the [Google](#) and [AirBnB](#) JavaScript Style Guides recommend.)



MSKU CENG

CENG-3507 Web Development & Programming

110

Variables best practices

- Use `const` whenever possible.
- If you need a variable to be reassignable, use `let`.
- **Don't use `var`.**

- You will
since `const`

- However
reason n

Aside: The internet has a **ton** of
misinformation about JavaScript!

Including several "accepted" StackOverflow answers,
tutorials, etc. Lots of stuff online is years out of date.

Tread carefully.

et with `var`

here's no

(This is also what the [Google](#) and [AirBnB](#) JavaScript Style Guides recommend.)



MSKU CENG

CENG-3507 Web Development & Programming

111

Types

JS **variables** do not have types, but the **values** do.

There are six primitive types ([mdn](#)):

- [Boolean](#): `true` and `false`
- [Number](#): everything is a double (no integers)
- [String](#): in `'single'` or `"double-quotes"`
- [Symbol](#): *(skipping this today)*
- [Null](#): `null`: a value meaning "this has no value"
- [Undefined](#): the value of a variable with no value assigned

There are also [Object](#) types, including `Array`, `Date`, `String` (the object wrapper for the primitive type), etc.



MSKU CENG

CENG-3507 Web Development & Programming

112

Numbers

```
const homework = 0.45;
const midterm = 0.2;
const final = 0.35;
const score =
    homework * 87 + midterm * 90 + final * 95;
console.log(score);    // 90.4
```



MSKU CENG

CENG-3507 Web Development & Programming

113

Numbers

```
const homework = 0.45;
const midterm = 0.2;
const final = 0.35;
const score =
    homework * 87 + midterm * 90 + final * 95;
console.log(score);    // 90.4
```

- All numbers are floating point real numbers. No integer type.
- Operators are like Java or C++.
- Precedence like Java or C++.
- A few special values: [NaN](#) (not-a-number), +Infinity, -Infinity
- There's a [Math](#) class: `Math.floor`, `Math.ceil`, etc.



MSKU CENG

CENG-3507 Web Development & Programming

114

Strings

```
let snack = 'coo';  
snack += 'kies';  
snack = snack.toUpperCase();  
console.log("I want " + snack);
```

**MSKU CENG**

CENG-3507 Web Development & Programming

115

Strings

```
let snack = 'coo';  
snack += 'kies';  
snack = snack.toUpperCase();  
console.log("I want " + snack);
```

- Can be defined with single or double quotes
 - Many [style guides](#) prefer single-quote, but there is no functionality difference
- Immutable
- No `char` type: letters are strings of length one
- Can use plus for concatenation
- Can check size via `length` property (not function)

**MSKU CENG**

CENG-3507 Web Development & Programming

116

Boolean

- There are two literal values for boolean: `true` and `false` that behave as you would expect
- Can use the usual boolean operators: `&&` `||` `!`

```
let isHungry = true;
let isTeenager = age > 12 && age < 20;

if (isHungry && isTeenager) {
  pizza++;
}
```



MSKU CENG

CENG-3507 Web Development & Programming

117

Boolean

- Non-boolean values can be used in control statements, which get converted to their "truthy" or "falsy" value:
 - `null`, `undefined`, `0`, `NaN`, `' '`, `''` evaluate to false
 - Everything else evaluates to true

```
if (username) {
  // username is defined
}
```



MSKU CENG

CENG-3507 Web Development & Programming

118

Equality

JavaScript's == and != are basically broken: they do an implicit type conversion before the comparison.

```
'' == '0' // false
'' == 0   // true
0 == '0'  // true
NaN == NaN // false
[''] == '' // true
false == undefined // false
false == null // false
null == undefined // true
```



MSKU CENG

CENG-3507 Web Development & Programming

119

Equality

Instead of fixing == and !=, the **ECMAScript standard** kept the existing behavior but added === and !==

```
'' === '0' // false
'' === 0   // false
0 === '0'  // false
NaN === NaN // still weirdly false
[''] === '' // false
false === undefined // false
false === null // false
null === undefined // false
```

Always use === and !== and don't use == or !=



MSKU CENG

CENG-3507 Web Development & Programming

120

Null and Undefined

What's the difference?

- `null` is a value representing the absence of a value, similar to `null` in Java and `nullptr` in C++.
- `undefined` is the value given to a variable that has not been a value.

```
let x = null;
let y;
console.log(x);
console.log(y);
```

<code>null</code>
<code>undefined</code>
>



MSKU CENG

CENG-3507 Web Development & Programming

121

Null and Undefined

What's the difference?

- `null` is a value representing the absence of a value, similar to `null` in Java and `nullptr` in C++.
- `undefined` is the value given to a variable that has not been a value.
 - ... however, you can also set a variable's value to `undefined`



```
let x = null;
let y = undefined;
console.log(x);
console.log(y);
```

<code>null</code>
<code>undefined</code>
>



MSKU CENG

CENG-3507 Web Development & Programming

122

Arrays

Arrays are Object types used to create lists of data.

```
// Creates an empty list
let list = [];
let groceries = ['milk', 'cocoa puffs'];
groceries[1] = 'kix';
```

- 0-based indexing
- Mutable
- Can check size via `length` property (not function)

**MSKU CENG**

CENG-3507 Web Development & Programming

123

Events

**MSKU CENG**

CENG-3507 Web Development & Programming

124

Event-driven programming

Most JavaScript written in the browser is **event-driven**: The code doesn't run right away, but it executes after some event fires.



Example:

Here is a UI element that the user can interact with.

Event-driven programming

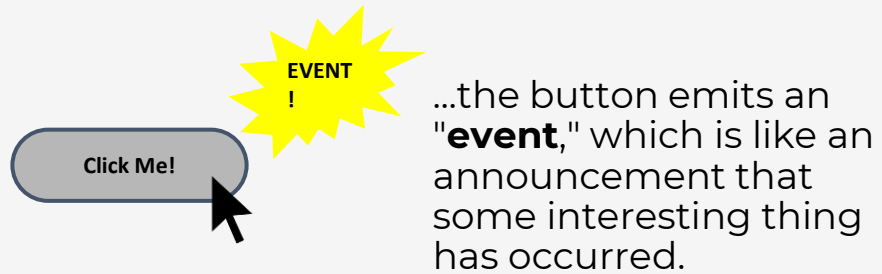
Most JavaScript written in the browser is **event-driven**: The code doesn't run right away, but it executes after some event fires.



When the user clicks the button...

Event-driven programming

Most JavaScript written in the browser is **event-driven**: The code doesn't run right away, but it executes after some event fires.



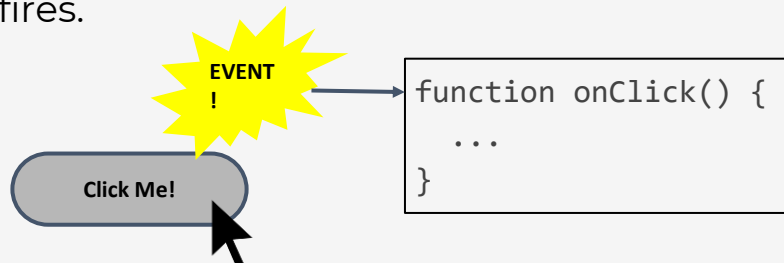
MSKU CENG

CENG-3507 Web Development & Programming

127

Event-driven programming

Most JavaScript written in the browser is **event-driven**: The code doesn't run right away, but it executes after some event fires.



Any function listening to that event now executes. This function is called an **"event handler."**



MSKU CENG

CENG-3507 Web Development & Programming

128

A few more HTML elements

Buttons:

```
<button>Click me</button>
```



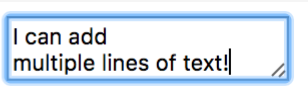
Single-line text input:

```
<input type="text" />
```



Multi-line text input:

```
<textarea></textarea>
```




MSKU CENG

CENG-3507 Web Development & Programming

129

Using event listeners

Let's print "Clicked" to the Web Console when the user clicks the given button:

```
<button>Click me</button>
```



We need to add an event listener to the button...

How do we talk to an element in HTML from JavaScript?



MSKU CENG

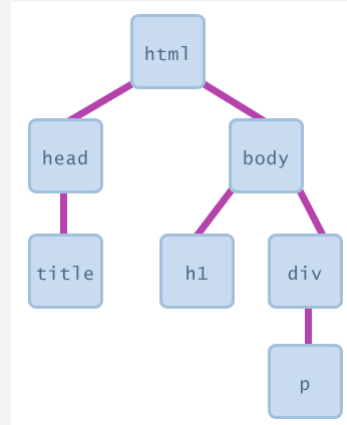
CENG-3507 Web Development & Programming

130

The DOM

Every element on a page is accessible in JavaScript through the **DOM: Document Object Model**

- The DOM is the tree of nodes corresponding to HTML elements on a page.
- Can modify, add and remove nodes on the DOM, which will modify, add, or remove the corresponding element on the page.



MSKU CENG

CENG-3507 Web Development & Programming

131

Getting DOM objects

We can access an HTML element's corresponding DOM object in JavaScript via the [querySelector](#) function:

```
document.querySelector('css selector');
```

- This returns the **first** element that matches the given CSS selector

```
// Returns the element with id="button"
```

```
let element = document.querySelector('#button');
```



MSKU CENG

CENG-3507 Web Development & Programming

132

Adding event listeners

Each DOM object has the following function:

`addEventListener(event name, function name);`

- **event name** is the string name of the [JavaScript event](#) you want to listen to
 - Common ones: click, focus, blur, etc
- **function name** is the name of the JavaScript function you want to execute when the event fires



MSKU CENG

CENG-3507 Web Development & Programming

133

```
<html>
  <head>
    <meta charset="utf-8">
    <title>First JS Example</title>
    <script src="script.js"></script>
  </head>
  <body>
    <button>Click Me!</button>
  </body>
</html>
```



```
function onClick() {
  console.log('clicked');
}

const button = document.querySelector('button');
button.addEventListener('click', onClick);
```



MSKU CENG

134



Error! Why?

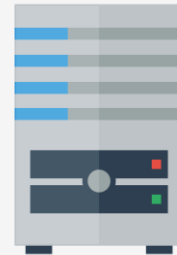
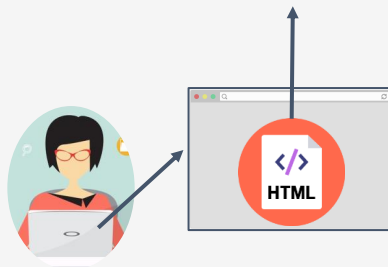


135

```

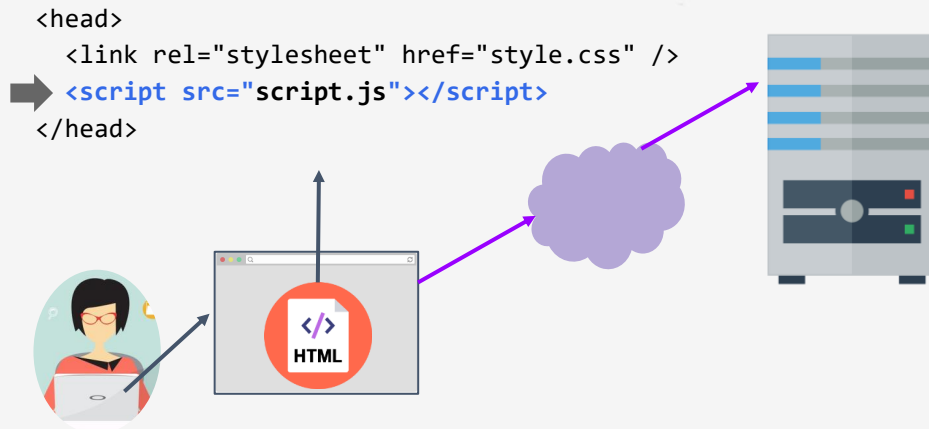
<head>
  <link rel="stylesheet" href="style.css" />
  <script src="script.js"></script>
</head>

```



CENG-3507 Web Development & Programming

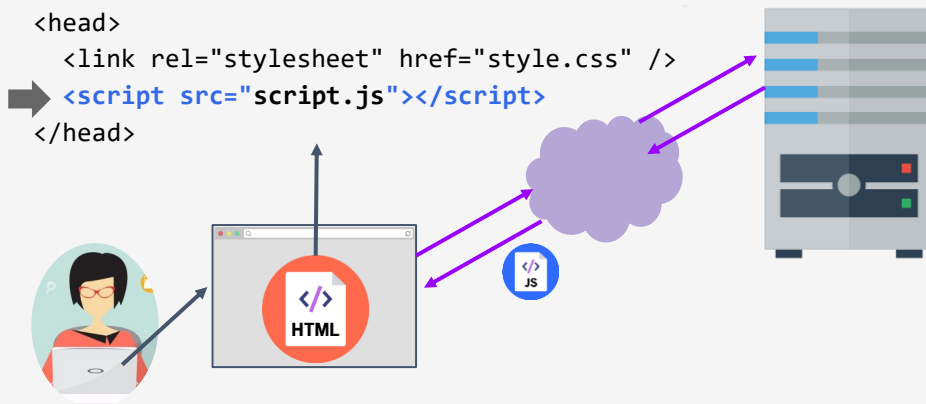
136



MSKU CENG

CENG-3507 Web Development & Programming

137

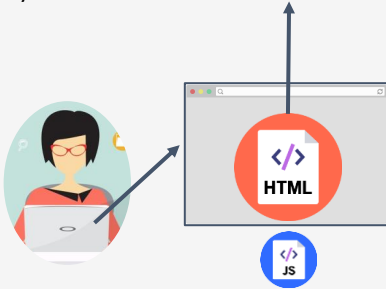


MSKU CENG

CENG-3507 Web Development & Programming

138

```
<head>
  <link rel="stylesheet" href="style.css" />
  ➔ <script src="script.js"></script>
</head>
```



```
function onClick() {
  console.log('clicked');
}

const button = document.querySelector('button');
button.addEventListener('click', onClick);
```

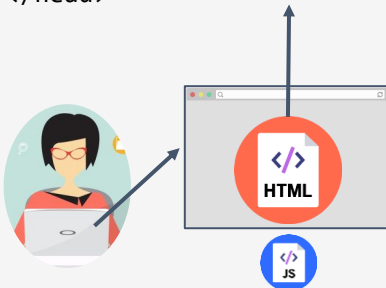


MSKU CENG

CENG-3507 Web Development & Programming

139

```
<head>
  <link rel="stylesheet" href="style.css" />
  ➔ <script src="script.js"></script>
</head>
```



```
function onClick() {
  console.log('clicked');
}

const button = document.querySelector('button');
button.addEventListener('click', onClick);
```

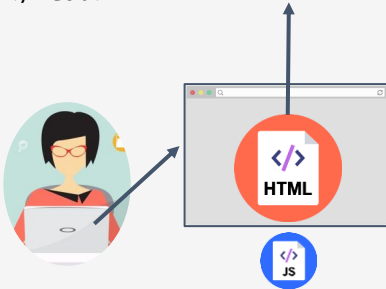


MSKU CENG

CENG-3507 Web Development & Programming

140

```
<head>
  <link rel="stylesheet" href="style.css" />
  ➡ <script src="script.js"></script>
</head>
```



```
function onClick() {
  console.log('clicked');
}

const button = document.querySelector('button');
button.addEventListener('click', onClick);
```

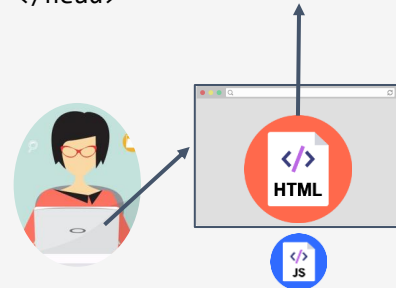
We are only at the `<script>` tag, which is at the top of the document... so the `<button>` isn't available yet.



CENG-3507 Web Development & Programming

141

```
<head>
  <link rel="stylesheet" href="style.css" />
  ➡ <script src="script.js"></script>
</head>
```



```
function onClick() {
  console.log('clicked');
}

const button = document.querySelector('button');
button.addEventListener('click', onClick);
```

Therefore `querySelector` returns `null`, and we can't call `addEventListener` on `null`.



CENG-3507 Web Development & Programming

142

Use defer

You can add the `defer` attribute onto the script tag so that the JavaScript doesn't execute until after the DOM is loaded ([mdn](#)):

```
<script src="script.js" defer></script>
```

**MSKU CENG**

CENG-3507 Web Development & Programming

143

Use defer

You can add the `defer` attribute onto the script tag so that the JavaScript doesn't execute until after the DOM is loaded ([mdn](#)):

```
<script src="script.js" defer></script>
```

Other old-school ways of doing this (**don't do these**):

- Put the `<script>` tag at the bottom of the page
- Listen for the "load" event on the window object

You will see tons of examples on the internet that do this. They are out of date. `defer` is [widely supported](#) and better.

**MSKU CENG**

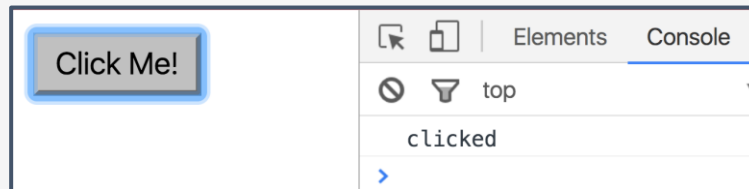
CENG-3507 Web Development & Programming

144


```
<html>
  <head>
    <meta charset="utf-8">
    <title>First JS Example</title>
    <script src="script.js" defer</script>
  </head>
  <body>
    <button>Click Me!</button>
  </body>
</html>
```

```
function onClick() {
  console.log('clicked');
}

const button = document.querySelector('button');
button.addEventListener('click', onClick);
```



145

Log messages aren't so interesting...

146

How do we interact with the page?



MSKU CENG

CENG-3507 Web Development & Programming

147

DOM object properties

You can access **attributes** of an HTML element via a property (field) of the DOM object

```
const image = document.querySelector('img');  
image.src = 'new-picture.png';
```

Some exceptions:

- Notably, you can't access the `class` attribute via `object.class`



MSKU CENG

CENG-3507 Web Development & Programming

148

Adding and removing classes

You can control **classes** applied to an HTML element via `classList.add` and `classList.remove`:

```
const image = document.querySelector('img');  
// Adds a CSS class called "active".  
image.classList.add('active');  
// Removes a CSS class called "hidden".  
image.classList.remove('hidden');
```

(More on <https://developer.mozilla.org/en-US/docs/Web/API/Element/classList>)

**MSKU CENG**

CENG-3507 Web Development & Programming

149

More next time!

**MSKU CENG**

CENG-3507 Web Development & Programming

150