

CS104

Javascript

lightweight interpreted

just in time compilation

- **document object model**

hierarchical representation
of HTML document]



converts page
into a tree

DOM

javascript

webpage

Structure, style, and content of the page can be changed dynamically.

- DOM essentially an API that provides a standardized interface to dynamically change the webpage

like a waiter in
a restaurant

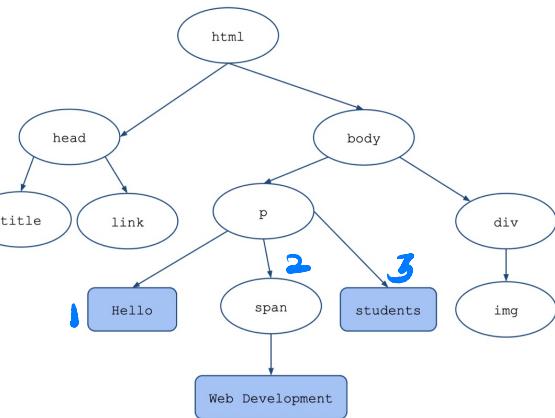
request - server - result

```

<html>
<head>
  <title>DOM Tutorial</title>
  <link rel="stylesheet"
    href="index.css">
</head>
<body>
  <p>
    1 Hello
    2 <span>Web Development</span>
    3 students
  </p>
  <div>
    
  </div>
</body>
</html>

```

Sample.html



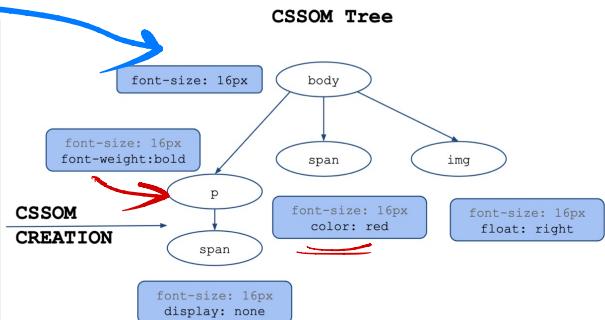
DOM Tree

```

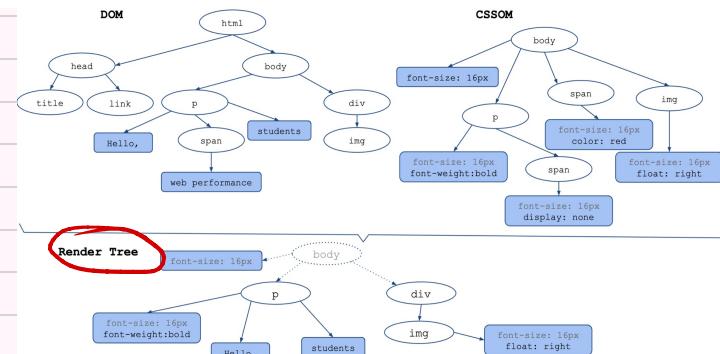
body {
  font-size: 16px;
}
p {
  font-weight:bold;
}
span {
  color: red;
}
p span {
  display: none;
}
img {
  float: right;
}

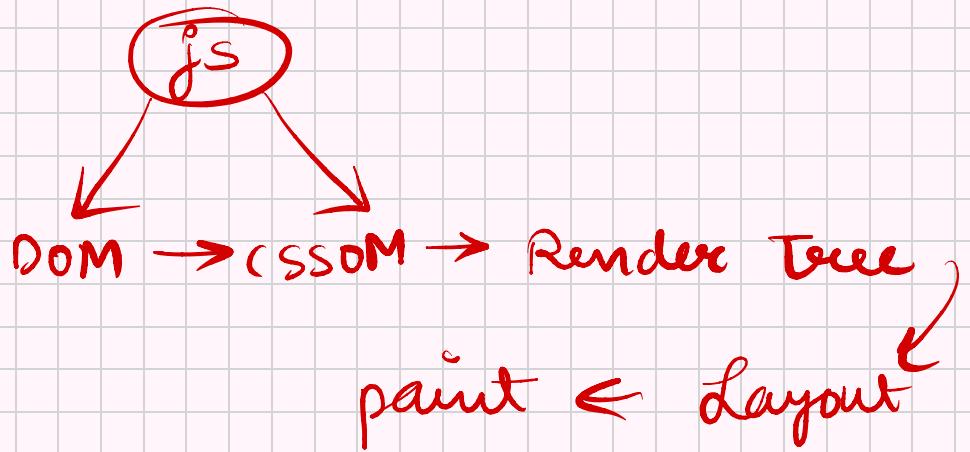
```

index.css



CSSOM Tree





- `console.log(typeof str1)`
will print var. type
- try catch works just like C++
- `text.replace(/\n/g,
)`
 - replaces all \n w/

 - all occurrences, "not just first"

Power of Javascript

```
<html>
<head>
  <title>JavaScript Example</title>
  <style>
    #message {
      color: blue;
    }
    .hidden {
      display: none;
    }
  </style>
</head>
<body>
  <h1 id="title">Original Title</h1>
  <button id="changeContentBtn">Change Content</button>
  <button id="changeStyleBtn">Change Style</button>
  <button id="hideElementBtn">Hide Element</button>
```

→ just CSS

remember that ids are unique

```
<p id="message">This is a message.</p>  
  
<form id="exampleForm">  
  <label for="email">Email: </label>  
  <input type="text" id="email" name="email"/>  
  <button type="submit">Submit</button>  
</form>  
  
<p id="errorMessage" class="hidden">Please enter a valid email address.</p>
```

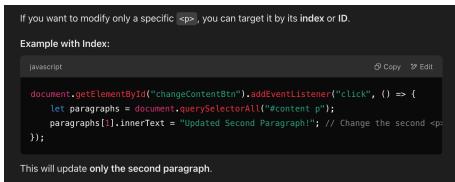
```
<script>
```

```
  // Change HTML Content
```

this is why
id is useful

```
  document.getElementById("changeContentBtn").addEventListener("click",  
  {  
    // entire HTML
```

```
    document.getElementById("title").innerText = "Updated Title!"  
  }) ;
```



gets the text
inside element

observes the element

could be
dblclick also
or many
more!

() =>

function
takes no
args

this is arrow func
regular:

"click" function() { }

```
// Change CSS Styling  
  
document.getElementById("changeStyleBtn").addEventListener("click", () => {  
  const message = document.getElementById("message");  
  message.style.color = "green";  
  message.style.fontSize = "20px";  
});
```

// Hide Element

```
document.getElementById("hideElementBtn").addEventListener("click", () => {  
  document.getElementById("message").classList.add("hidden");  
});
```

★ hides it
(Internal
CSS)

.match()
matches arg w/ all elements
returns first match by default

```
// Validate Form

document.getElementById("exampleForm").addEventListener("submit", (event) => {
  event.preventDefault();
  const emailInput = document.getElementById("email").value;
  const errorMessage = document.getElementById("errorMessage");

  if (!emailInput.includes("@")) {
    errorMessage.classList.remove("hidden");
  } else {
    errorMessage.classList.add("hidden");
    alert("Form submitted successfully!");
  }
});

</script>
</body>
</html>
```

validate
entering
email

Original Title

[Change Content](#)

[Change Style](#)

[Hide Element](#)

This is a message.

Email:

[Submit](#)

Updated Title!

[Change Content](#)

[Change Style](#)

[Hide Element](#)

This is a message.

Email:

kq

[Submit](#)

Please enter a valid email address.

javascript has C-style syntax

<script> JS code </script>

Display ➔

JavaScript can "display" data in different ways:

- Writing into an HTML element, using innerHTML edits contents of ref. element
- Writing into the HTML output using document.write() rewrite the HTML completely
- Writing into an alert box, using window.alert() pop up
- Writing into the browser console, using console.log() ✓

Variables in JavaScript

gets scoped to
nearest function block → not if or loop
otherwise global

Var

not used now
causes bugs
w/ scopes

let

only if
reassignment
needed

scope just
like normal
C++ variables

const

use by
default

auto

ok but
best to declare

- global variable

may cause
data type issues

Operators

very similar to C++

== v/s ===

↳ only value

$$2^3 = 2 * 2 * 2$$

↳ value & type

Datatypes in JavaScript

String `() []` `' '` `` ``

need ↗

doesn't need ↗

Boolean T / F

template like ↗

Number 623

Undefined let `x`

Big int 62...37n

Symbol `x=Symbol('α')`

`x` can't be reassigned

Null on purpose

Object Array

Arrays

join()

join all elements
into a string
separated by
any of join

push & pop like
vector

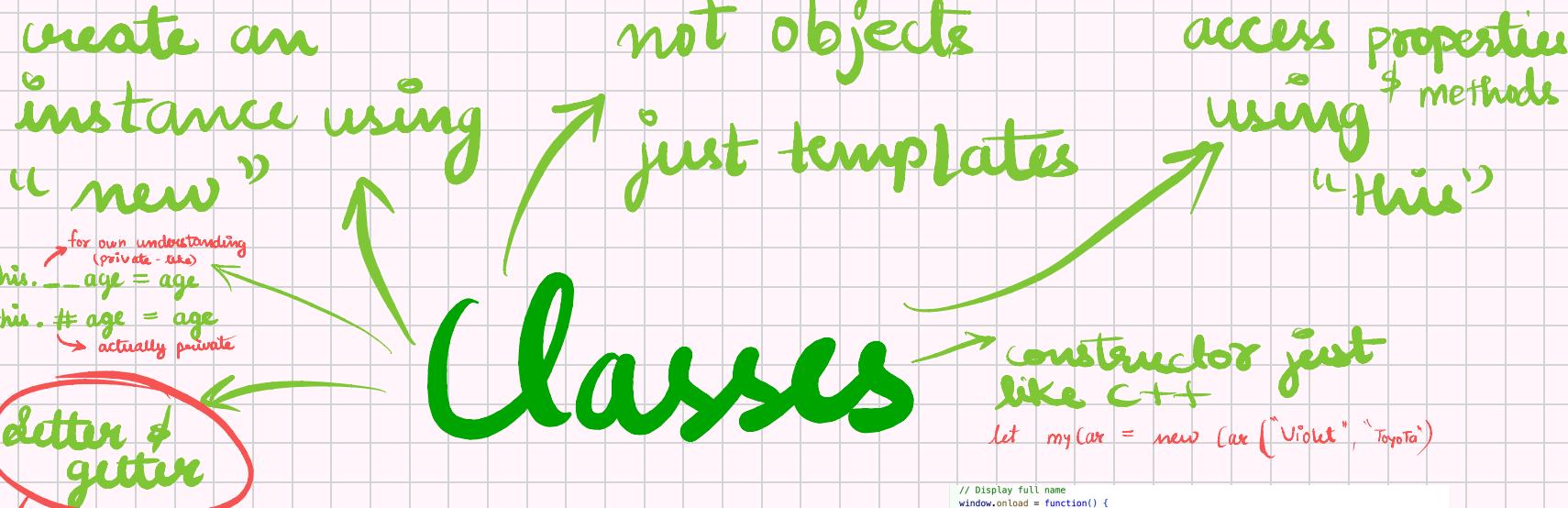
sort()
alphabetically
reverse()

property
name : value pairs
accessed via obj name . property

also variables can contain many values & methods

declare with "new"
let str1 = new Number(10)

Objects



```
// Display full name
window.onload = function() {
  document.getElementById('output').innerHTML += '<p>Full Name: ${person1.fullName}</p>';

  // Display age using getter
  document.getElementById('output').innerHTML += '<p>Age of Person1: ${person1.age}</p>';

  // Set new age using setter
  person1.age = 30;
  document.getElementById('output').innerHTML += '<p>Updated Age: ${person1.age}</p>';

  // Set an invalid age (negative)
  person1.age = -5;
}

// Call the static method
Person.greet();

// Display age using getter
document.getElementById('output').innerHTML += '<p>Age of Person2: ${person2.age}</p>';

}
```

mean literal template
\${ } allows you to add any JS expression inside it

Difference?
• Getter (()) behaves like a property + (2.16)
• Normal function requires parentheses + (2.16)

get

for set:
Difference?
• Function plus requires the argument + (2.16)
• Normal function requires a method call + (2.16)

Info dump P

use `rtfind` for
whole string

```
size_t pos = path.find_last_of('/');
if (pos == std::string::npos) {
    directoryName = root->getName();
    fileName = path;
} else {
    directoryName = path.substr(0, pos);
    fileName = path.substr(pos);
}
```

→ finds last occurrence
if ("like")

matches any character
not entire word

d if dir (l = symbolic link)

permissions

```
-rw-r--r-- 12 linuxize users 12.0K Apr  8 20:51 filename.txt
| [ - ] [ - ] [ - ]- [-----] [---]
| | | | | | | |
| | | | | | +-----> 7. Group
| | | | | | +-----> 6. Owner
| | | | +-----> 5. Alternate Access Method
| | | +-----> 4. Others Permissions
| | +-----> 3. Group Permissions
| +-----> 2. Owner Permissions
+-----> 1. File Type
```

The first character shows the file type. It can be a regular file (-), directory (d), a symbolic link (l), or any other special type of file.

UNIX INFO DVMP

grep

-F == -R

-A [num]

-B

-C

"

"

+5

-5

±5

} doesn't
work w)

-O

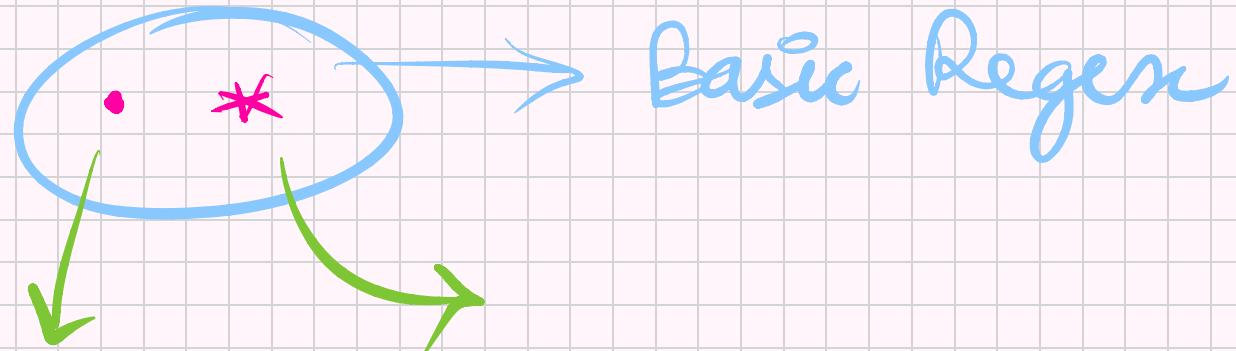
grep
"u/n"
doesn't
work

subsequences
wol (not wolf)

only matched
part

grep -v → Invert match

grep -E → extended Reger



any
single
char

WC → --byte = -c → no of bytes

char → -m

\m | word { byte

ASCII

1 byte = 1 char

emoji = n bytes

Cut -d

-f

field / key

delim.

Sort -t