JSON

EXAMPLE 15: CREATE AN OBJECT THE "JSON" WAY

RECAP

EXAMPLE 15: CREATE AN OBJECT THE "JSON" WAY

"JSON" = JavaScript Object Notation

JSON IS SIMPLY A HANDY WAY TO SPECIFY HOW OBJECTS LOOK USING ONLY TEXT- THINK OF IT AS A COMPETITOR TO XML, OR EVEN HTML :-) TECHNICALLY

RECAP

EXAMPLE 15: CREATE AN OBJECT THE "JSON" WAY

"JSON" = JavaScript Object Notation

JSON IS SIMPLY A HANDY WAY TO SPECIFY HOW OBJECTS LOOK USING ONLY TEXT

RECAP

EXAMPLE 15: CREATE AN OBJECT THE "JSON" WAY

"JSON" = JavaScript Object Notation

JSON IS

A COMPETITOR

RECAP

EXAMPLE 15: CREATE AN OBJECT THE "JSON" WAY

"JSON" = JavaScript Object Notation

JSON IS A

DATA-INTERCHANGE FORMAT

RECAP

EXAMPLE 15: CREATE AN OBJECT THE "JSON" WAY

"JSON" = JavaScript Object Notation

```
<script>


var rectangle = {
  length : 5.0,
  breadth : 3.5,
  color : 'Red'
};

function objectStuff() {
  console.log("Our rectangle has length = " + rectangle.length);
  console.log("Our rectangle has breadth = " + rectangle.breadth);
  console.log("Our rectangle has area = "+ rectangle.color);
}
```

JSON IS SIMPLY A HANDY WAY TO SPECIFY HOW OBJECTS LOOK USING ONLY TEXT- THINK OF IT AS A COMPETITOR TO XML, OR EVEN HTML :-) TECHNICALLY

RECAP

EXAMPLE 19: CREATE AN OBJECT THE "JSON" WAY

"JSON" = JavaScript Object Notation

```
var rectangle = {
    length: 5.0,
    breadth : 3.5,
    color : 'Red'
};


function objectStuff() {
    console.log("our
```

RECAP

EXAMPLE 19: CREATE AN OBJECT THE "JSON" WAY

"JSON" = JavaScript Object Notation

```
var rectangle = {
    length: 5.0,
    breadth : 3.5,
    color : 'Red'
};


function objectStuff() {
    console.log("our
```

RECAP

EXAMPLE 19: CREATE AN OBJECT THE "JSON" WAY

"JSON" = JavaScript Object Notation

DECLARE THE OBJECT, LIKE YOU WOULD ANY OTHER VARIABLE

```
var rectangle = {
    length : 5.0,
    breadth : 3.5,
    color : "Red"
};
```

RECAP

```
function objectStuff() {
    console.log("our
```

var rectangle

EXAMPLE 19: CREATE AN OBJECT THE "JSON" WAY

"JSON" = JavaScript Object Notation

THE NAME OF THE OBJECT

```
};
```

```
function objectStuff() {
  console.log("our
```

RECAP

EXAMPLE 15: CREATE AN OBJECT THE "JSON" WAY

"JSON" = JavaScript Object Notation

```
var rectangle = {
  length: 5.0,
  breadth : 3.5,
  color : 'Red'
};
```

DEFINE THE OBJECT BETWEEN
A PAIR OF CURLY BRACES

RECAP

```
function objectStuff() {
  console.log("our
```

EXAMPLE 15: CREATE AN OBJECT THE "JSON" WAY

"JSON" = JavaScript Object Notation

```
length: 5.0,
breadth : 3.5,
color : 'Red'
```

};

THE PROPERTIES FOLLOW

```
function objectStuff() {
  console.log("our
```

RECAP

EXAMPLE 15: CREATE AN OBJECT THE "JSON" WAY

"JSON" = JavaScript Object Notation

```
length: 5.0,
breadth : 3.5,
color : 'Red'
};
```

EACH PROPERTY IS A KEY-VALUE PAIR

RECAP

EXAMPLE 15: CREATE AN OBJECT THE "JSON" WAY

"JSON" = JavaScript Object Notation

```
length: 5.0,
breadth : 3.5,
color : 'Red'
```

};

EACH PROPERTY IS A KEY-VALUE PAIR

RECAP

EXAMPLE 15: CREATE AN OBJECT THE "JSON" WAY

"JSON" = JavaScript Object Notation

```
length: 5.0,
breadth : 3.5,
color : 'Red'
```

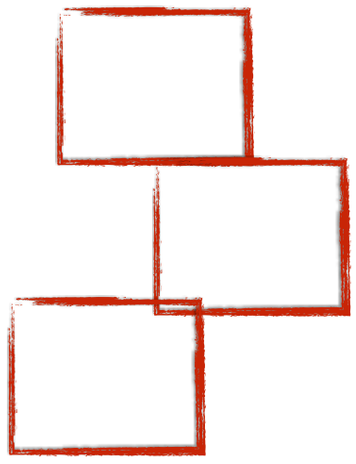
EACH PROPERTY IS A KEY-VALUE PAIR

RECAP

EXAMPLE 15: CREATE AN OBJECT THE "JSON" WAY

"JSON" = JavaScript Object Notation

```
length : 5.0,
breadth : 3.5,
color : 'Red'
```



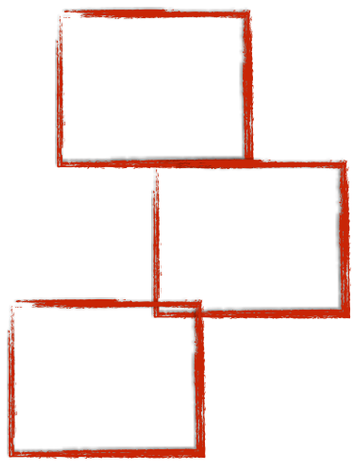
THE KEY AND VALUE ARE SEPARATED BY A COLON

RECAP

EXAMPLE 15: CREATE AN OBJECT THE "JSON" WAY

"JSON" = JavaScript Object Notation

```
length: 5.0,
breadth : 3.5,
color : 'Red'
};
```



ALL PROPERTIES EXCEPT THE LAST ARE FOLLOWED BY A COMMA

RECAP

EXAMPLE 15: CREATE AN OBJECT THE "JSON" WAY

"JSON" = JavaScript Object Notation

```
var rectangle = {
  length : 5.0,
  breadth : 3.5,
  color : 'Red'
};
```

TO THE OUTSIDE WORLD, AN OBJECT IS SIMPLY A VARIABLE (GLOBAL, IN THIS CASE)

```
function objectStuff() {
  console.log("our rectangle has length = " +
rectangle.length);
  console.log("our rectangle has breadth = " +
rectangle.breadth);
  console.log("our rectangle has area = " +
rectangle.color);

}
```

RECAP

EXAMPLE 15: CREATE AN OBJECT THE "JSON" WAY

"JSON" = JavaScript Object Notation

```
var rectangle = {
  length : 5.0,
  breadth : 3.5,
  color : 'Red'
};


function objectStuff() {
  console.log("our rectangle has length = " +
  rectangle.length);
  console.log("our rectangle has breadth = " +
  rectangle.breadth);
  console.log("our rectangle has area = " +
  rectangle.color);

}
```

```
our rectangle has length = 5
our rectangle has breadth = 3.5
our rectangle has area = Red
```

THE PROPERTIES CAN BE ACCESSED VIA A DOT...

RECAP

WEB BROWSERS AND WEB SERVERS

YOU WAKE UP IN THE MORNING AND WANT TO READ THE NEWSPAPER

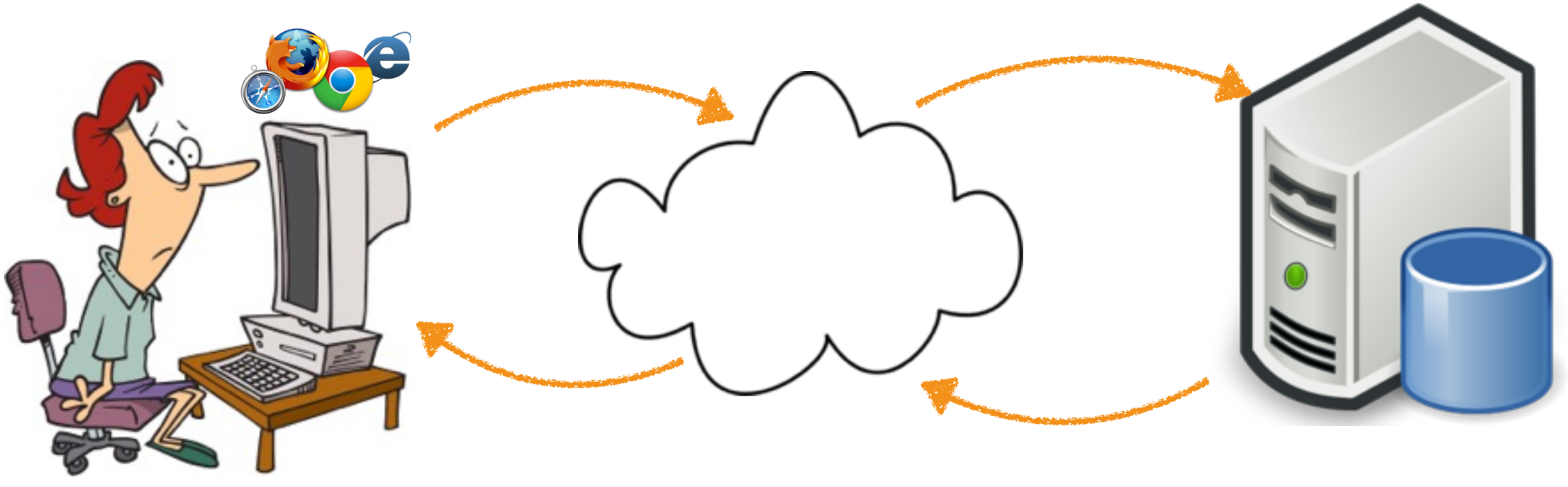
YOU OPEN UP YOUR FAVORITE BROWSER

AND TYPE
WWW.NYTIMES.COM

WEB BROWSERS AND WEB SERVERS

REQUEST

WEB SERVERS OF



RESPONSE = WEB PAGE
(HTML + CSS + JAVASCRIPT)

RECAP

THE WEB SERVER AT WWW.NYTIMES.COM WILL SEND BACK INFORMATION

TEXT
FORMATTING
IMAGES
VIDEOS...

THIS INFORMATION IS IN A FORM THE BROWSER UNDERSTANDS

RESPONSE = WEB PAGE
(HTML + CSS + JAVASCRIPT)

RESOURCES

HTML
CSS
JAVASCRIPT
IMAGES, VIDEOS...

RECAP

RESPONSE = WEB PAGE (HTML + CSS + JAVASCRIPT)

THE BROWSER KNOWS HOW TO INTERPRET HTML, CSS AND JAVASCRIPT

RECAP

THE BROWSER KNOWS HOW TO INTERPRET HTML, CSS AND JAVASCRIPT

AND

RENDER THE WEBPAGE

RECAP

THE BROWSER KNOWS HOW TO INTERPRET HTML, CSS AND JAVASCRIPT

AND RENDER THE WEBPAGE

THE BROWSER KNOWS HOW TO INTERPRET HTML, CSS AND JAVASCRIPT

AND RENDER THE WEBPAGE

BROWSER

WEBPAGE

HTML

CSS

JAVASCRIPT

RECAP

THE BROWSER KNOWS HOW TO INTERPRET HTML, CSS AND JAVASCRIPT AND RENDER THE WEBPAGE

BROWSER

WEBPAGE

- HTML
- CSS
- JAVASCRIPT

- HTML
  - HEAD
  - BODY
    - H1
    - P

THIS TREE IS CALLED THE DOM

"DOCUMENT-OBJECT-MODEL"

RECAP

IN ALL OF THIS, WE ENTIRELY GLOSSED OVER THE **WEB SERVER**

THE BROWSER KNOWS **HOW TO INTERPRET AND RENDER THE**

BROWSER

WEBPAGE

- HTML
- CSS
- JAVASC

- HTML
  - HEAD
  - BODY
    - H1
    - P

THIS TREE IS CALLED "DOCUMENT-OBJECT-

WEB SERVER

JAVASCRIPT, HTML AND CSS ALL WORK ON THE CLIENT'S BROWSER

BUT ULTIMATELY, THERE IS A LOT THAT YOU CANNOT DO ON THE CLIENT AND WHAT CAN'T BE DONE ON THE CLIENT MUST BE DONE

ON THE SERVER

AND WHAT CAN'T BE DONE ON THE CLIENT MUST BE DONE

ON THE SERVER

HEAVY-DUTY STUFF TENDS TO BE DONE ON THE SERVER -

LOOKING UP USER INFORMATION

UPDATING PAYMENT INFORMATION

MAINTAINING A USER'S SHOPPING CART

AND LOTS AND LOTS OF STUFF

HEAVY-DUTY STUFF TENDS TO BE DONE ON THE SERVER

THE BROWSER KNOWS HOW TO INTERPRET AND RENDER THE

BROWSER

WEBPAGE

- HTML
- CSS
- JAVASC

- HTML
  - HEAD
  - BODY
    - H1
    - P

THIS TREE IS CALLED "DOCUMENT-OBJECT-

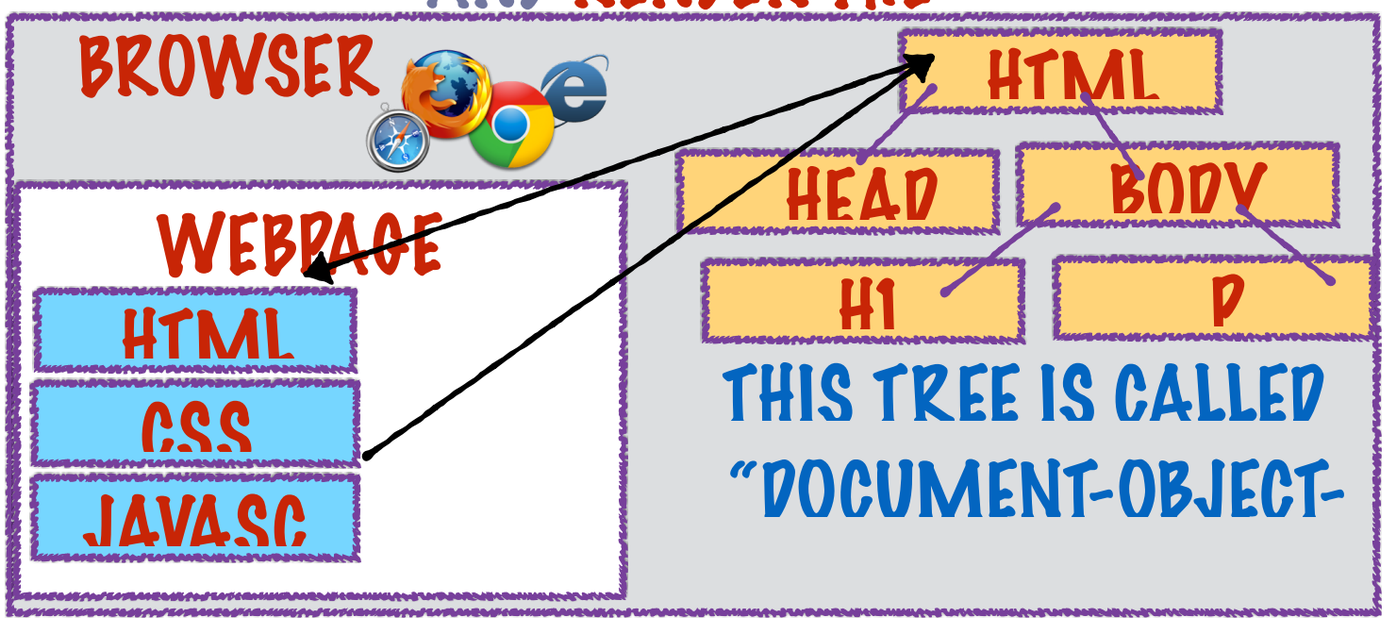
SERVER-SIDE CODE IS USUALLY THE BULK - THE HEART OF LARGE-SCALE WEB APPLICATIONS

AND SERVER-SIDE CODE IS ALMOST NEVER IN JAVASCRIPT

AND SERVER-SIDE CODE IS ALMOST NEVER IN JAVASCRIPT

HEAVY-DUTY STUFF

THE BROWSER KNOWS HOW TO INTERPRET AND RENDER THE

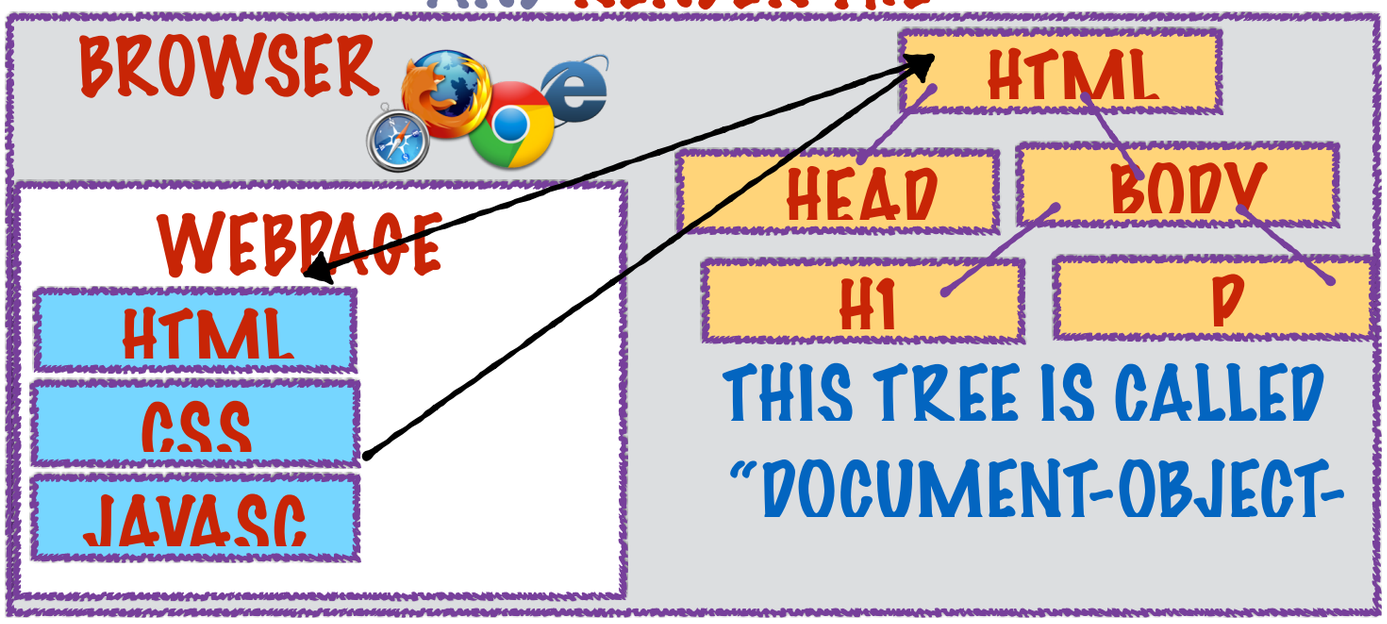


AND SERVER-SIDE CODE IS
ALMOST NEVER IN JAVASCRIPT

JAVA, C#, C++, RUBY,...

HEAVY-DUTY STUFF

THE BROWSER KNOWS HOW TO INTERPRET
AND RENDER THE



SERVER-SIDE CODE IS ALMOST NEVER IN JAVASCRIPT

JAVA, C#, C++, RUBY,...

BUT IT DOESN'T MATTER, BECAUSE THE SERVER CAN ALWAYS COMMUNICATE WITH THE JAVASCRIPT USING **JSON**

JSON

JSON IS A MAGICAL WAY IN WHICH

A JAVASCRIPT OBJECT CAN BE

CREATED FROM A STRING

THE SERVER CODE SIMPLY SENDS JSON STRINGS

JSON

JSON IS A MAGICAL WAY IN WHICH

A JAVASCRIPT OBJECT CAN BE CREATED FROM A STRING

THE SERVER CODE SIMPLY SENDS JSON STRINGS

AND THE BROWSER PASSES THESE STRINGS TO THE JAVASCRIPT IN A WEBPAGE

JSON

JSON IS A MAGICAL WAY IN WHICH

A JAVASCRIPT OBJECT CAN BE CREATED FROM A STRING

THE SERVER CODE SIMPLY SENDS JSON STRINGS AND THE BROWSER PASSES THESE STRINGS TO THE JAVASCRIPT IN A WEBPAGE

JAVASCRIPT CONVERTS THESE STRINGS TO OBJECTS!

USING JSON THE SERVER CAN CREATE VERY VERY COMPLEX OBJECTS ON THE CLIENT!

USING JSON THE SERVER
CAN CREATE VERY VERY
COMPLEX OBJECTS ON THE CLIENT!

THE BROWSER KNOWS HOW TO INTERPRET
AND RENDER THE