COMP4021
Internet Computing

A First Look at JavaScript

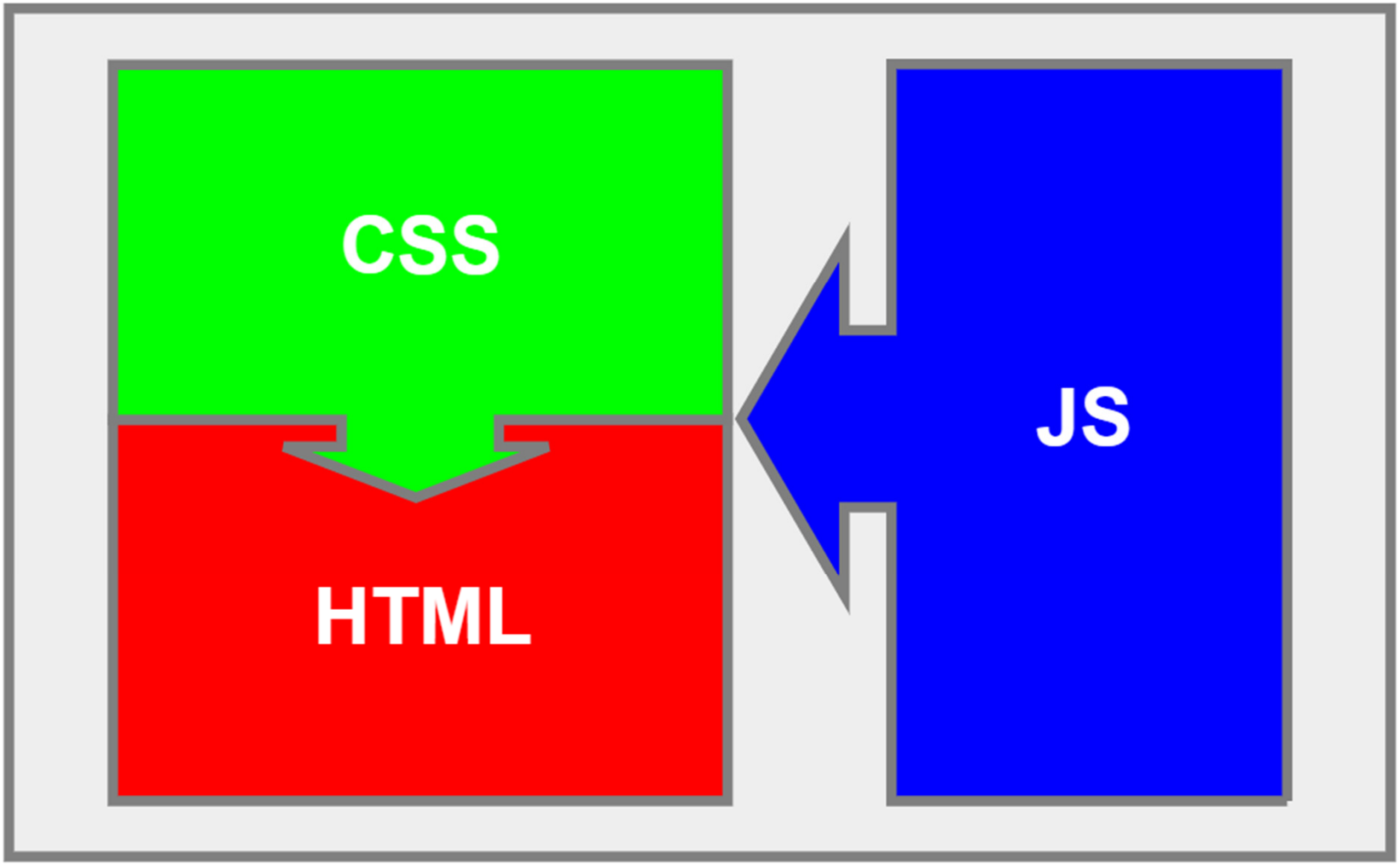
David Rossiter & Gibson Lam

JavaScript

- JavaScript (JS) is a scripting language used inside a browser (also other places)
- Just to be clear: although they have a similar name, JavaScript and Java are two completely different languages

basic idea: JS can change anything at anytime

Main Browser Components



Putting JavaScript in Webpages

- JavaScript can go anywhere inside the HTML page, e.g. inside the head section or the body section
- It has to be inside `<script> … </script>`
- Here we use JavaScript to show a small window which has a message when the HTML page is loaded in a browser

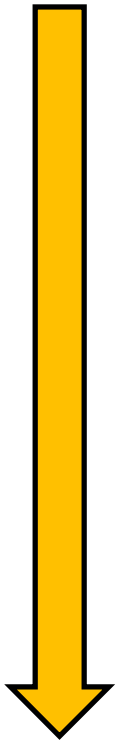
```
<!DOCTYPE html>
<html>
<head>
    ...
    <script>
    alert("How are you?");
    </script>
</head>
<body>
    ...
</body>
</html>
```

The Order of Running JavaScript

- When you put multiple JavaScript code inside an HTML file, the execution order follows the order that they are inside the file
- This happens no matter where you put the `<script>` tag, i.e. in the head or body section of the file

```
<!DOCTYPE html>
<html>
<head>
    ...
    <script>...</script>
    ...
    <script>...</script>
    ...
</head>
<body>
    ...
    <script>...</script>
    ...
    <script>...</script>
    ...
</body>
</html>
```

Execution order



Asking for Text Input

- A simple way to ask for text input is using `prompt()` which shows a small window for entering text

- Creating a variable

```
<script>
var name = prompt("What is your name?");
alert("Hi! " + name + ".");
</script>
```

- Concatenation

Basic Events

- Previous examples use JS that runs while the page is loading
- It is often useful to start JS when a particular *event* occurs
- A typical way to write code for an event is to:
  1. Create a function containing the JS code that you want to run when the event happens
  2. Assign the function to the event
- We will demonstrate two simple events:
  - The load event
  - The click event

A Load Event Example

- In this example the alert window is shown, after the page has loaded

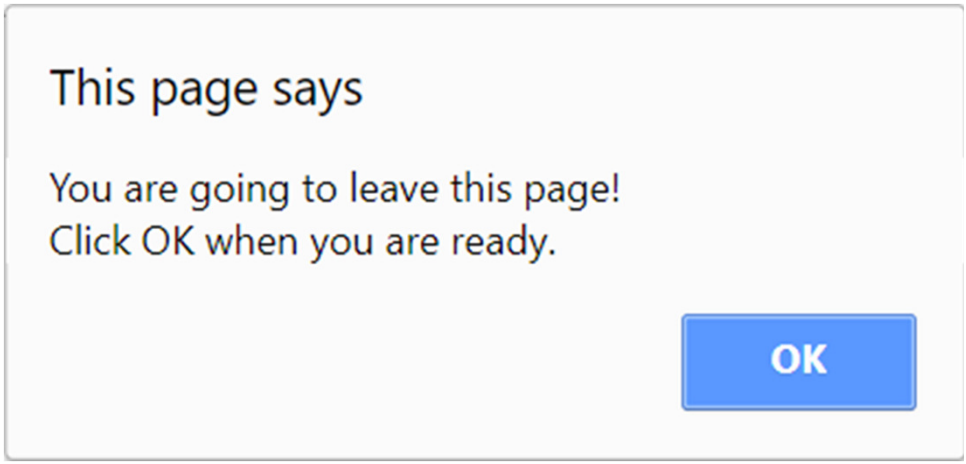
```
<!DOCTYPE html>
<html>
<head>
    ...
    <script>
    function show() {
        alert("How are you?");
    }
    </script>
</head>
<body onload="show()" >
    ...
</body>
</html
```

Run show() when the load event occurs

A Click Event Example

- \n means 'go to next line'

```
<script>
function warning() {
    alert("You are going to leave this page!\nClick OK
                                    when you are ready.");
}
</script>
...
<a onclick="warning()"
  href="https://www.cse.ust.hk">
  CSE Department</a>
```



alert 自带的OK按钮

- When the link is clicked, the message will be shown
- After clicking OK the next page is loaded

A Quick Note About Debugging

- Chrome DevTools has a full set of debugging tools
  - You can set breakpoints, pause execution, and so on
- When you develop your JavaScript code you can use `alert()` to create messages for yourself
- However, a better way is to use `console.log()`
- It shows whatever is generated in the console window
- Regular users of the web page won't see it

```
console.log("About to execute code");
```