The Browser Technology

By: Laams Innovation Lab

Lecture Contents

Decomposition Principle

Understanding Internet and the WWW

Understanding the relationship between HTML, CSS & JavaScript

The Browser Components

Using the Chrome Development Tools

Using HTML, CSS & JavaScript to Manipulate the Browser

Decomposition Principle

Decomposition principle refers to **dividing** up an **object** to *separate* **parts**, or **combining** *separate* **parts** to form a **whole**.

When decomposing think of *fixed* or *dynamic* parts, *parts* holding *constituent parts*, *lifetime* of the *parts* compared to the *whole*, and *parts* which are *shared*.

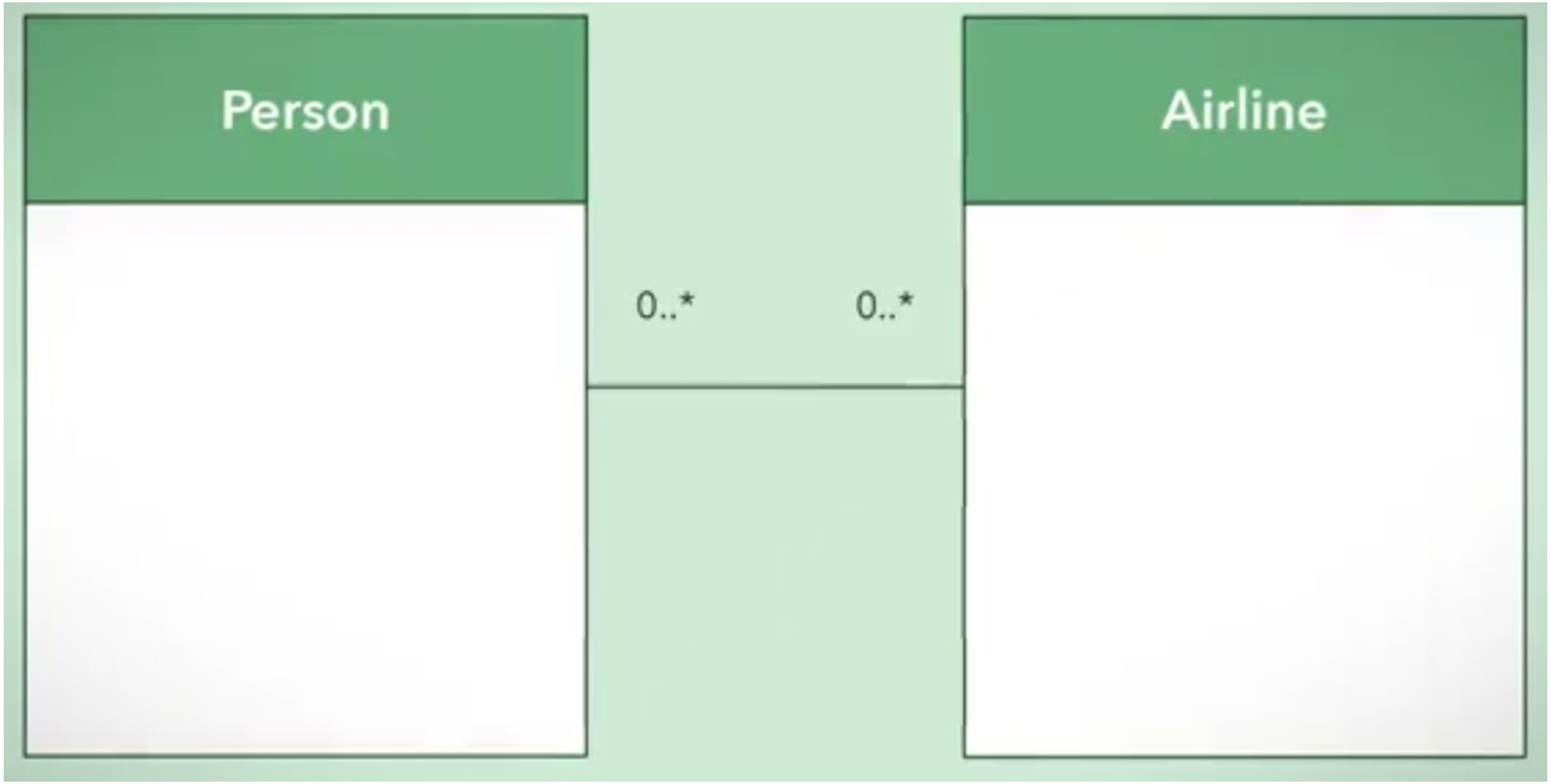
Type of Decomposition Relationships

There are three types of relationship in decomposition, which define the relationship between the whole and the part and are as following: **association**, **aggregation**, **composition**.

Association Relationship

Association is “**has some**” relationship, which indicates that there is a **loose** and **timely** relationship between **independent** **objects**.

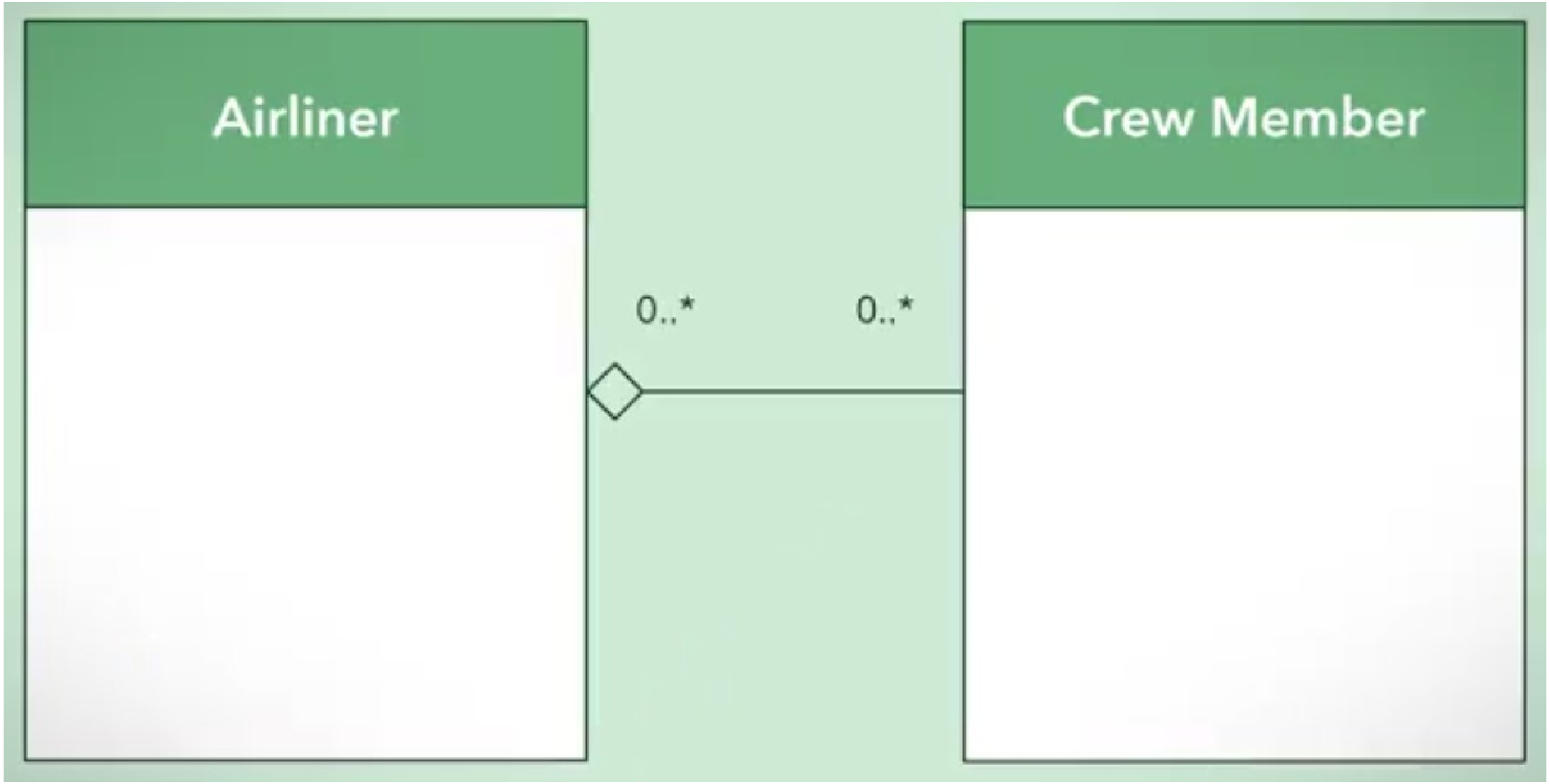
Association UML Class Diagram



Aggregation Relationship

Aggregation is “**has a**” relationship, indicating that the **whole** has **parts** which belong to it; however, the relationships is considered weak, and the **whole** and **part** can exist independently.

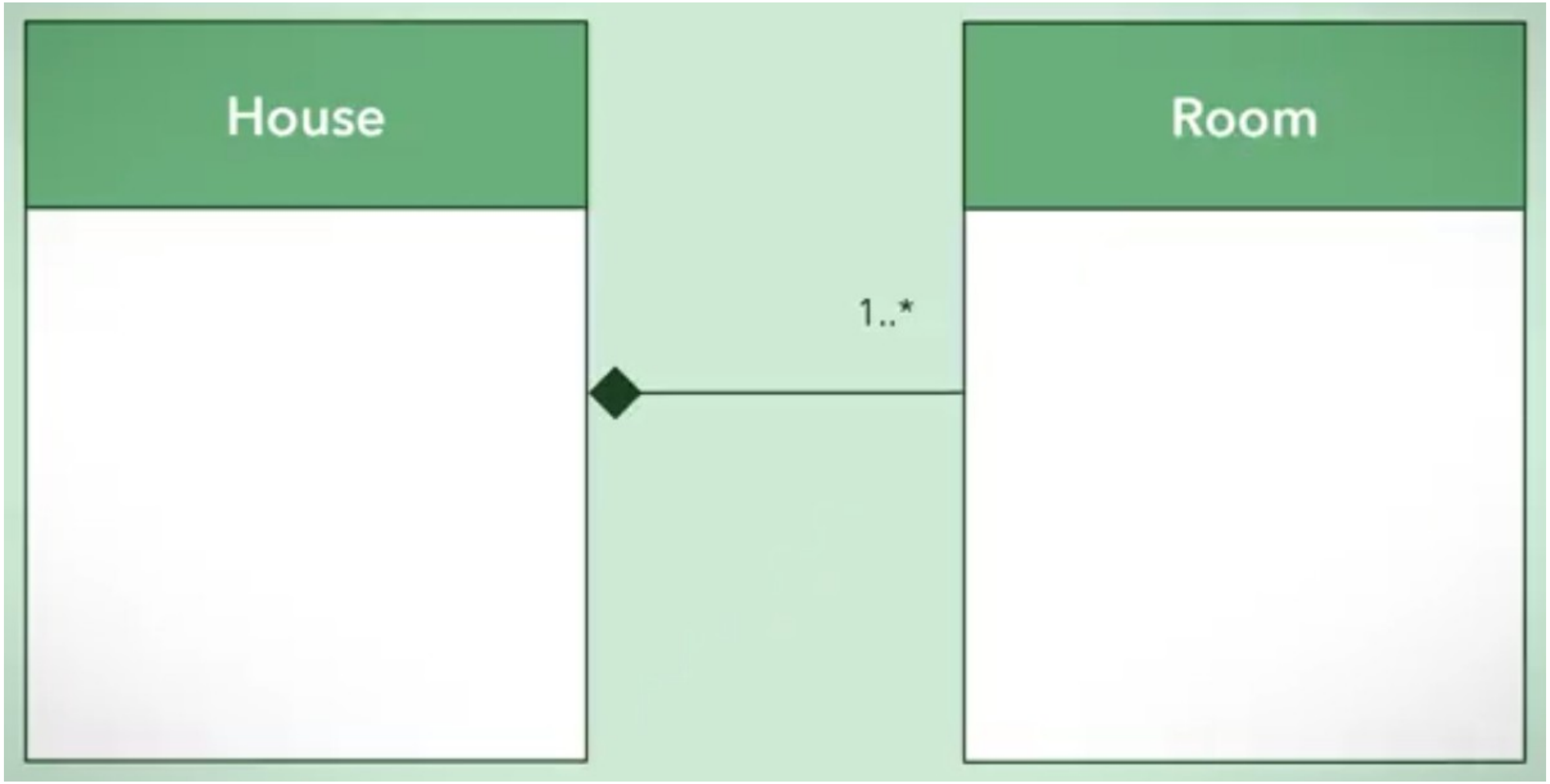
Aggregation UML Class Diagram



Composition Relationship

Composition is "**has a strong**" relationship, which indicates the **whole** **cannot exist** without the **part**. If whole is **destroyed** then parts is also **destroyed**. You usually access **parts** through the **whole**.

Composition UML Class Diagram



Benefits of Decomposition Principle

Decomposition helps us **break down** the problem into **smaller pieces**, to better understand and work with the smaller problems at a time. It makes code **re-usability** possible by sharing parts. It also makes the code **flexible**, as you can change each part independently. Also, with all of the aforementioned benefits our code becomes more **maintainable**.

What is the internet & who created it?

What is World Wide Web & who created it?

What is the World Wide Web Consortium (W3C) and its purpose?

What is URI, URL & URN, and how are they related to IP?

What is browser?

How does the Search Engines work?

What is the Internet?

The internet is a **distributed** **global** **system** of interconnected computer **networks**, which is made up of **independently** operated **networks** with **no central control**, and uses **TCP/IP** protocol for **transmission**.

Internet was co-created by Vint Cerf and Bob Kahn based on the **ARPANET** project in early 1970s.

Who is Sir Tim-Berners Lee?

Tim-Berners Lee, a British Software Engineer, is known for being the **inventor** of the **World Wide Wide**. He proposed a **global hypertext document system** which would be **using** the **internet** to share and transmit research related **information**.

His proposal became a success in 1991, when he **developed** the first **web server**, the first **web client** (browser), and the worlds first **website**; hence, the **World Wide Web** was born.

He then **founded** the **World Wide Web Consortium** at MIT for standardization of all the thing which is on the web.

What is WWW?

World Wide Web (WWW), also referred to as the **web**, is a **client-server global hypertext**-based **information system** which uses **Hypertext Transfer Protocol** (**HTTP**) to share documents (usually in the form of **HTML**) over the internet, and has been one of the driving force of information revolution.

What is W3C?

World Wide Web Consortium (W3C), founded by Tim-Berners Lee, creates **standardization** for the web, making the web **accessible** to all, and is in charge of standardizing **HTML** and **CSS** (previously **JavaScript** and **DOM**).

What is the DOM?

Document Object Model (DOM) is a **standard** for **representing** and **manipulating** the logical **tree** structure of a **document** in memory. It refers to **JavaScript objects**, whose **nodes** are created parallel to **elements** in a document. DOM though being JavaScript objects are not standardized by the JavaScript Language.

It has multiple interfaces, such as **Document** (represents HTML DOM), **XMLDocument** (represents XML DOM), The **Window** (represents the browser window), and numerous others…

The HTML DOM Tree

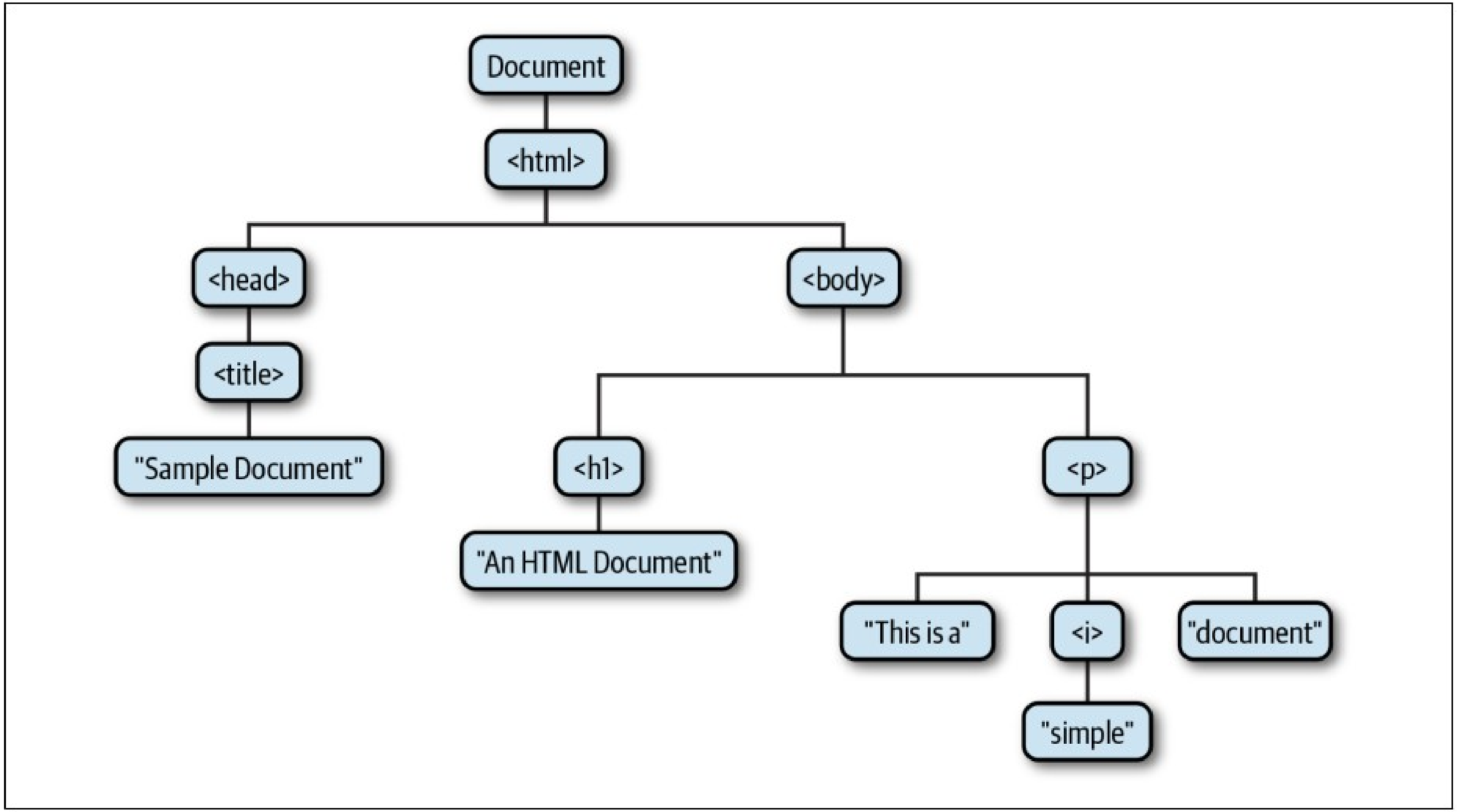


Figure 15-1. The tree representation of an HTML document

What is HTML?

Hypertext Markup Language is used to **structure** and give **meaning** to a web document content.

What is CSS?

Cascading Style Sheet is a language of style rules used to apply **styling** and **layout** to HTML content.

What is JavaScript?

JavaScript is a general purpose programming language, primarily used as a scripting language on the web to make the web pages dynamic.

What is Web Browser?

World Wide Web is a hypertext-based information system built on top of the internet.

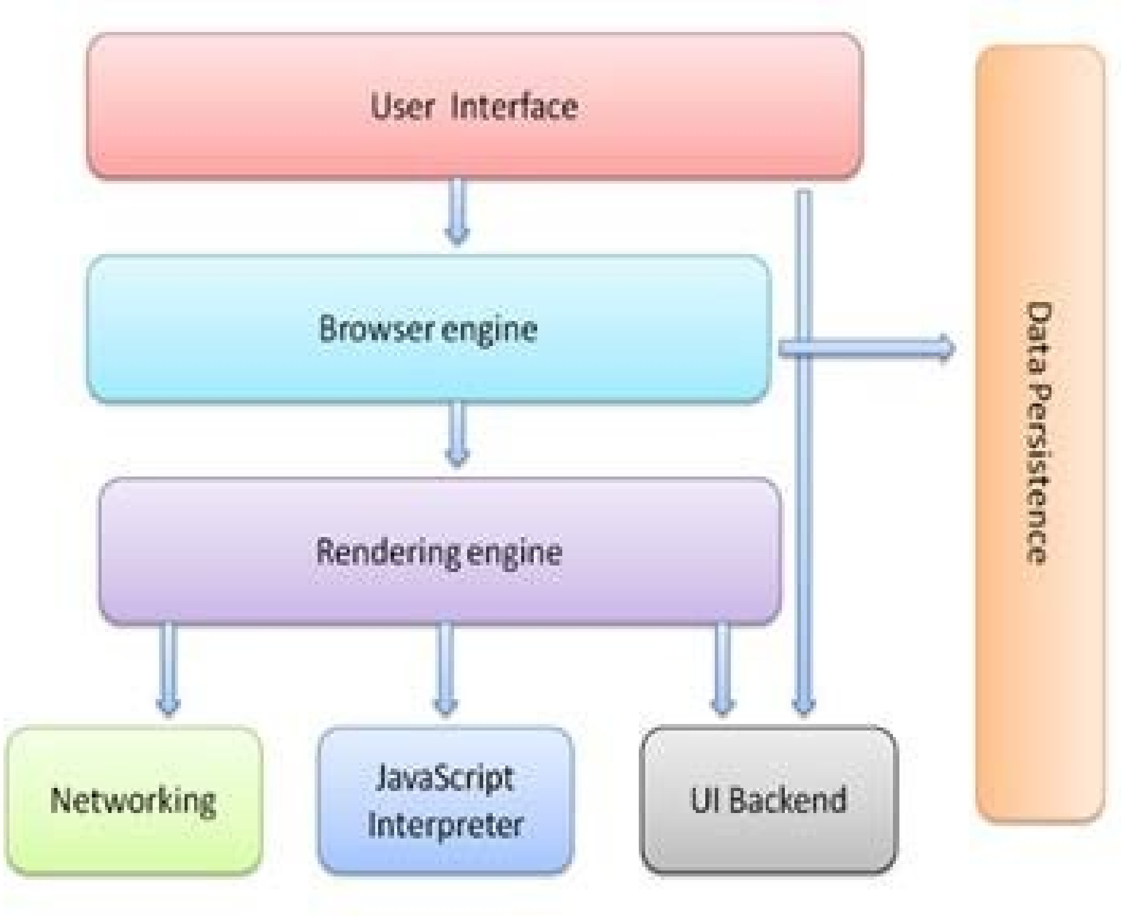
The modern browsers are equipped with manipulable parts, such as, the DOM, CSSOM, JavaScript Interpreter, debugging capabilities, cookies, storage systems (databases, such as IndexedDB, and WebSQL) memory management, network information, and so many more...

What is Web Browser?

World Wide Web is a hypertext-based information system built on top of the internet.

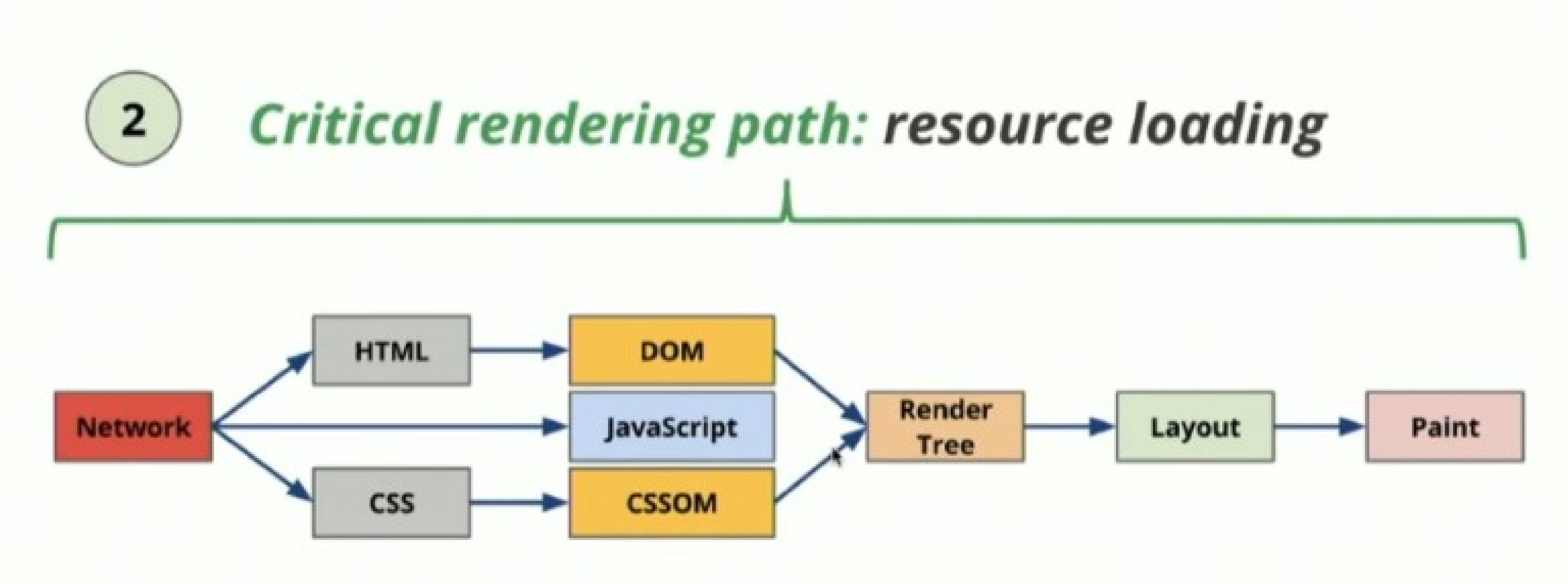
The modern browsers are equipped with manipulable parts, such as, the DOM, CSSOM, JavaScript Interpreter, debugging capabilities, cookies, storage systems (databases, such as IndexedDB, and WebSQL) memory management, network information, and so many more...

Browser's Components



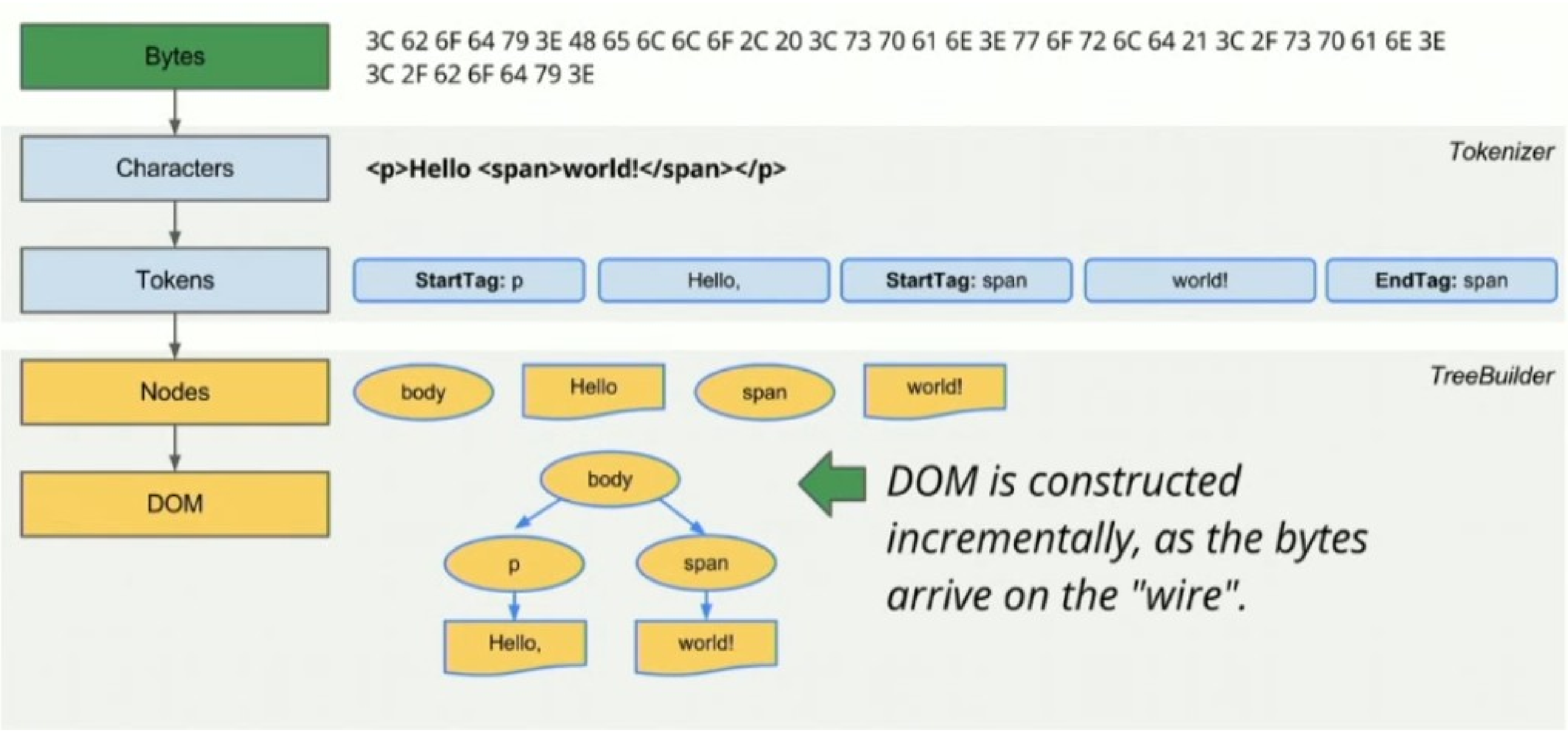
Experimenting with HTML, CSS, JavaScript

Browser’s Render Flow



2 *Critical rendering path: resource loading*

HTML Parsing



Bytes

3C 62 6F 64 79 3E 48 65 6C 6C 6F 2C 20 3C 73 70 61 6E 3E 77 6F 72 6C 64 21 3C 2F 73 70 61 6E 3E
3C 2F 62 6F 64 79 3E

Characters

Tokenizer

<p>Hello <span>world!</span></p>

Tokens

StartTag: p | Hello, | **StartTag:** span | world! | **EndTag:** span

Nodes

TreeBuilder

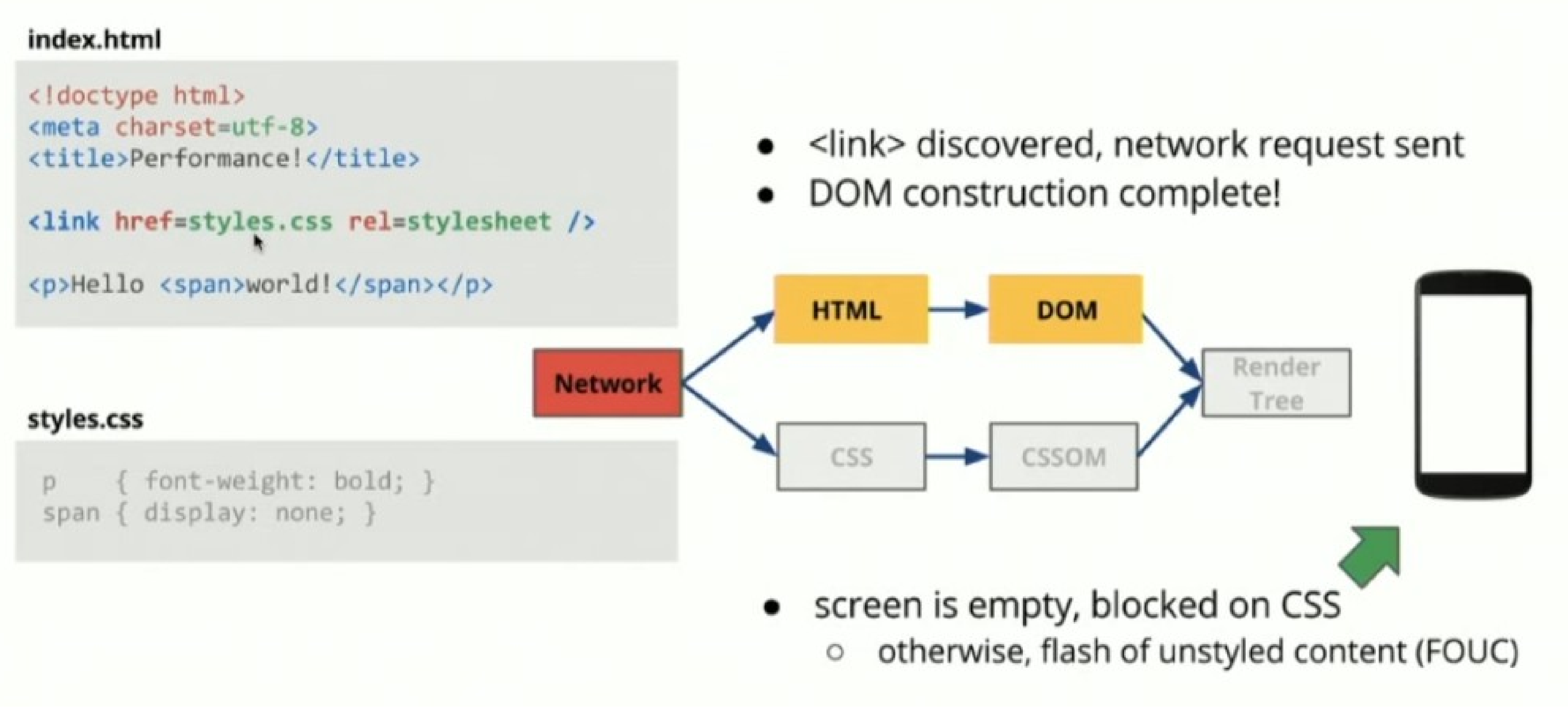
body | Hello | span | world!

DOM

body
p
span
Hello,
world!

DOM is constructed incrementally, as the bytes arrive on the "wire".

HTML DOM Rendering



index.html

```
<!doctype html>
<meta charset=utf-8>
<title>Performance!</title>

<link href=styles.css rel=stylesheet />

<p>Hello <span>world!</span></p>
```

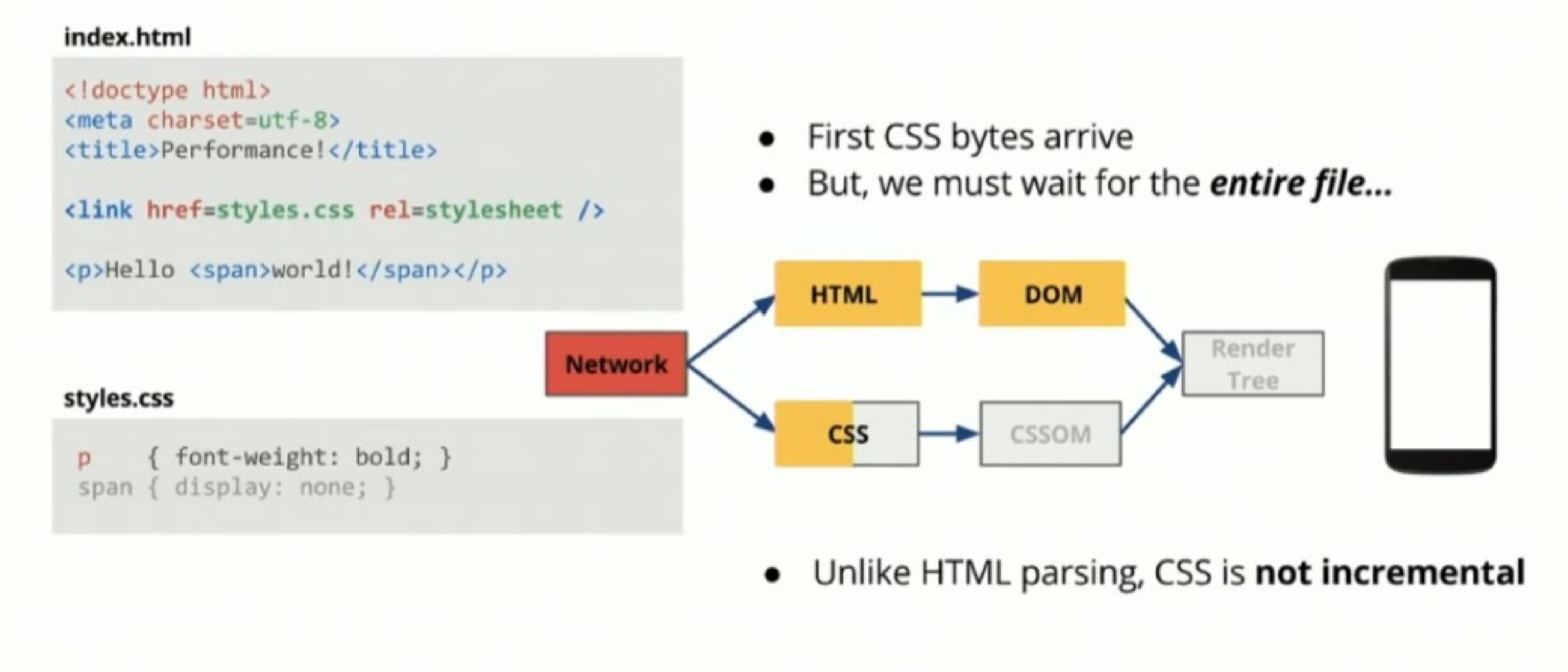
styles.css

```
p    { font-weight: bold; }
span { display: none; }
```

- <link> discovered, network request sent
- DOM construction complete!

- screen is empty, blocked on CSS
    - otherwise, flash of unstyled content (FOUC)

Waiting CSS Bytes to Arrive:



index.html

```
<!doctype html>
<meta charset=utf-8>
<title>Performance!</title>

<link href=styles.css rel=stylesheet />

<p>Hello <span>world!</span></p>
```

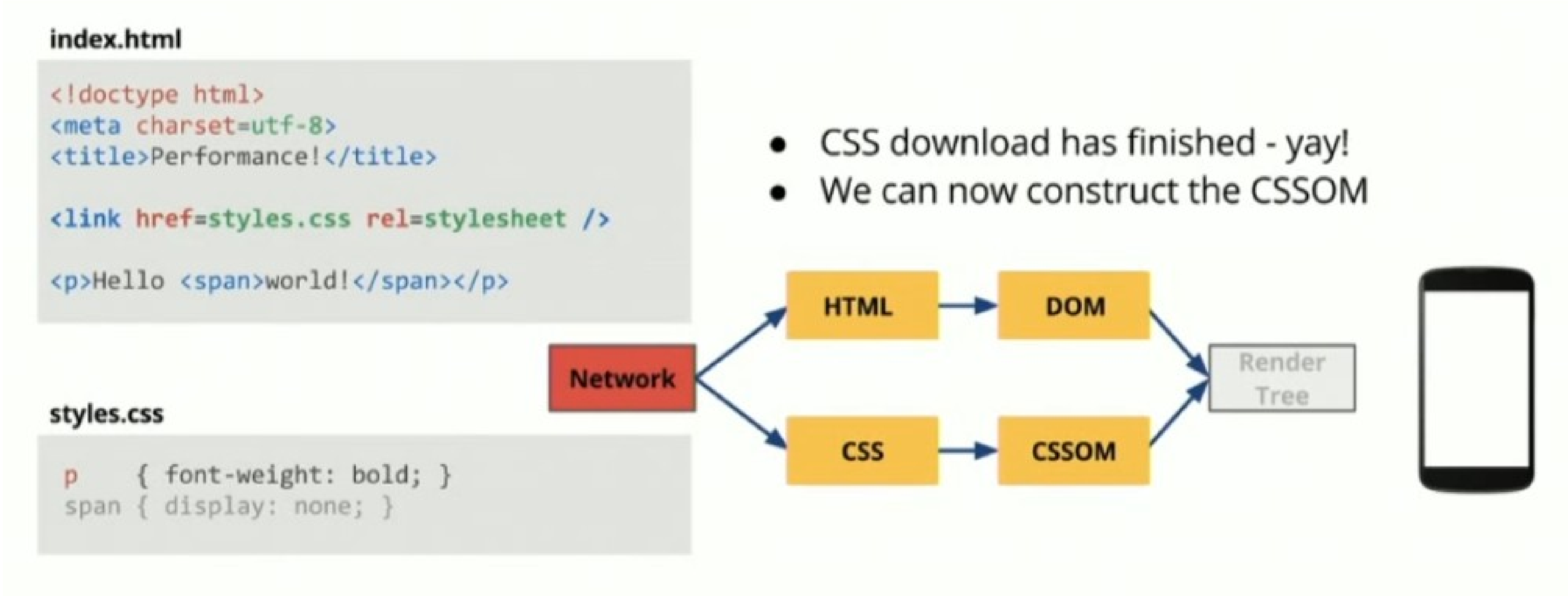
styles.css

```
p    { font-weight: bold; }
span { display: none; }
```

- First CSS bytes arrive
- But, we must wait for the ***entire file...***

- Unlike HTML parsing, CSS is **not incremental**

CSSOM constructed:



index.html

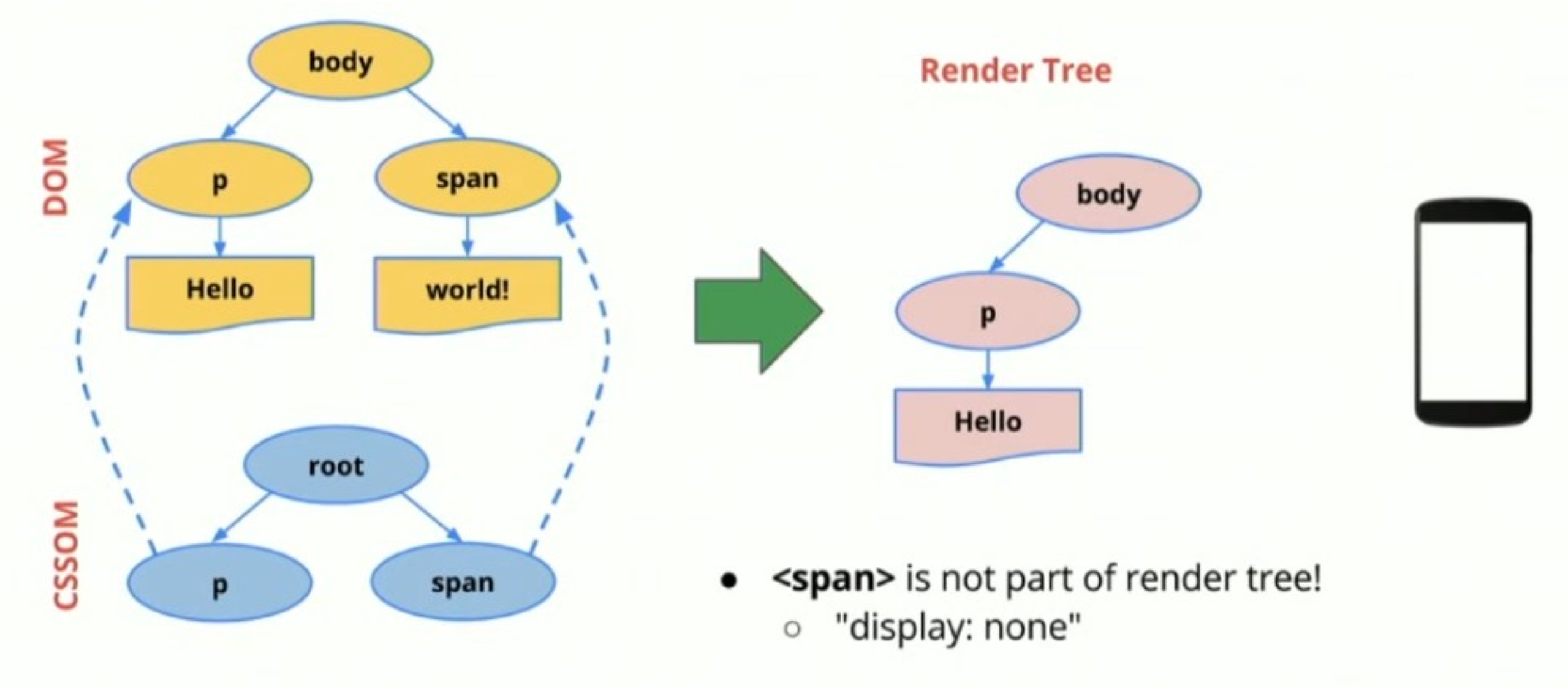
```
<!doctype html>
<meta charset=utf-8>
<title>Performance!</title>

<link href=styles.css rel=stylesheet />

<p>Hello <span>world!</span></p>
```

styles.css

```
p    { font-weight: bold; }
span { display: none; }
```

- CSS download has finished - yay!
- We can now construct the CSSOM

Interpolated HTML & CSS Tree:



- **<span>** is not part of render tree!
  - "display: none"