Week 6

HTTP

Request Response protocol(协议)
Client ask(request) to server and wait, the server response related to that request.

HTTP Request
GET /index.html HTTP/1.1
Host: www.bristol.ac.uk host that connect to
Connection: close

HTTP Response
HTTP/1.1 200 OK
Content-Type: text/html;
charset=UTF-8
Content-Length:1009

<!DOCTYPE html>
…..(follow the document they request)

Response codes
1xx: information
2xx: success (200 OK, 201 Created, …)
3xx: redirect (301 Moved Permanently, …) (the file has been move to another path and server will send the location back)
4xx: **client** error (400 Bad request, 403 Forbidden, 404 Not Found, …)
5xx: **server** error (500 Internal Server Error, …)

Content-Type (first priority for the browser)
Content-Type: text/html; charset=UTF-8
Explain the format body
Charset explain the encoding of web page
text/plain, text/html, ….
Image/jpeg, application/pdf, video/mp4….

Methods
GET
HEAD ask header request (check how large the file it is)
POST/PUT, DELETE

Cookie
Stateless protocol

Cookie protocol
The server will remember something from client.

Internet

Network Stack
HTTP
TLS
TCP
IP
Computer architecture stuff

IP: Internet Protocol
TCP: Transmission Control Protocol

TCP ports
80 HTTP
443 TLS (more secure communication) (build on top of TCP)
22 SSH
8000, 8080, … development (unofficial)

URL

URL parts

https://bristol.ac.uk/engineering/departments
scheme: how interpret the rest of URL.
host: the server wanna connect to/ address request.
path: specific resource on a machine/ specific for the GET request
query (question mark '?'): use for pass parameter to the resources, separate by '&' for each parameter

Rest

Representational State transfer
FACT: HTTP is stateless

- State goes in the request
- Resources have names (URLS)
- Use HTTP verbs as intended

GET /files/README.txt
GET /files?name=README.txt

HTML5

h1 …. h6 headings(with different size)
<html> wrap all content on page
<head> will not appear in the web page, but include key information that appear in search engine.
<meta> Represents metadata that cannot be represented by other HTML meta-related elements, like <base>, <link>, <script>, <style> or <title>
<title> will appear in the tab of a browser
<p> paragraphs

new line
 unordered list
 ordered list
 list item
<section> To structure your content
 Generic inline tag
 Generic block tab
<!-- --> comments
<Strong> mark up the content like: **mark up**
<i> use to convey a meaning traditionally conveyed by italics like: *italics*
<b> use to convey a meaning traditionally conveyed by bold like: **bold**
<u> use to convey a meaning traditionally conveyed by underline like: underline
<dl> stands for "description list".
<dt> is used to indicate the term being defined or described within a description list.
<dd> provides the details or description of the term specified by <dt>.
<hr /> the thematic break like:

Ron was backed into a corner by the marauding netherbeasts. Scared, but determined to protect his friends, he raised his wand and prepared to do battle, hoping that his distress call had made it through.

Meanwhile, Harry was sitting at home, staring at his royalty statement and pondering when the next spin off series would come out, when an enchanted distress letter flew through his window and landed in his lap. He read it hazily and sighed; "better get back to work then", he mused

<footer>
<nav> navigation bar
<aside> sidebar, often place inside <main>

Replacement:

< with <
> with >
“ with "
‘ with '
& with &

Links

Our Courses

Bristol.ac.uk/students/info.html :
“/courses” => Bristol.ac.uk/courses
“courses” => Bristol.ac.uk/student/courses
“../courses” => Bristol.ac.uk/courses

Form

```
<form method=”post” action=”/comment”>
    <p>
        <label for=”name”>Name:</label>
        <input id=”name” name=”name” />
    </p>
    <p>
        <button type =”submit”>OK</button>
    </p>
</form>
```

Validation:
<input required type=”number”>
Autocomplete
<input type="text”
Autocomplete=”name”>

Select:

```
<select name=”animal”>
    <option value=”dog”>Dog</option>
    <option value=”cat”>Cat</option>
</select>
```

Table:

```
<table>
    <thead>
```

```
    <tr><th>Name</th><th>ID</th></tr>
    </thead>
    <tbody>
        <tr><td>Sarah</td><td>100</td></tr>
        <tr><td>John</td><td>101</td></tr>
    </tbody>
</table>
```

<tr> table row
<td> table data

Week 7

CSS

Selector

p <p>
.important   ‘.’ Use for class
#title   ‘#’ use for id

.container p select all <p> under class=”container”
.container > p select direct children <p> after class="container

```
<div class="container">
    <p>This paragraph is selected because it is a direct child of .container.</p>
    <div>
        <p>This paragraph is not selected </p>
    </div>
</div>
```

.container ~ p

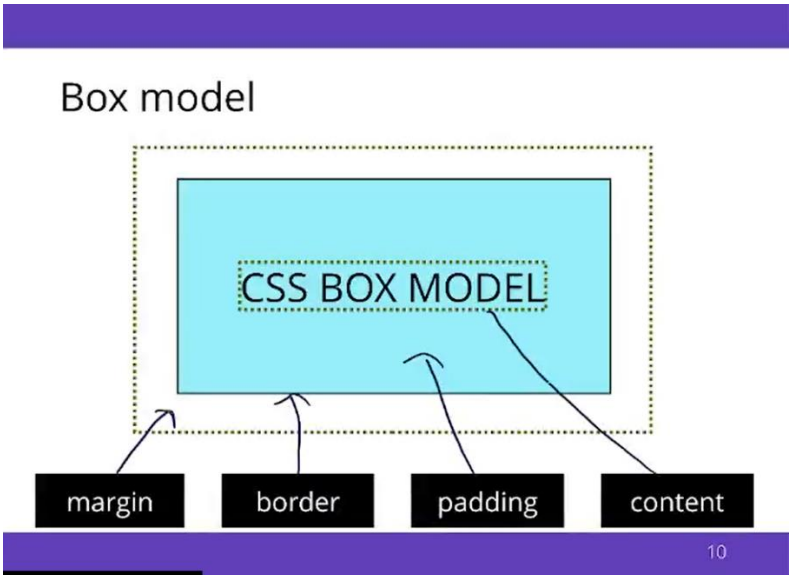
.container + p select immediately follows elements

```
<div>
    <div class="container">Example</div>
    <p>This paragraph is selected because it immediately follows .container.</p>
    <p>This paragraph is not selected, even though it follows .container</p>
</div>
```

Values

color: red;
background-color: #ff0000;
border-color: rgba(255, 0, 0, 1.0);(前三位为 rgb 数值，最后一位为透明度，1.0 最高，0 最低)

Box Model



margin-top: 10px;
padding-left: 20px;
margin: 10px 5px 15px 20px; (top, right, bottom, left)
border: 1px solid black;

Units of measurement

Dpi(ppi)

Dots(points)per inch

(avoid to use these absolute value)
px　pixel
pt　point
cm, in, ….

em　width of letter m (父系的大小 如 html 为 14px，则子系的 header 如果为 1.5em，则等效于 14*1.5 =　() px
ex　height of letter x

lh line height (+ space)

vw, vh 1% of viewport width/ height

rem root em(html) （根的相对大小，即<html>）

% parent width/height

Example

```
html { font-size: 12pt; }
p    { font-size: 1rem; }
h1 {
    font-size: 1.8rem;
    margin-top: 2ex;
    margin-bottom: 1ex;
}
h2   { font-size: 1.4 rem; }
```

```
html, body {
        margin: 0; (use 0 then all units are same e.g. 0pt = 0px….)
        padding: 0;
}
h1 {
    background-color: yellow;
    width: 100%;
}
```

CSS grid

```
.container {
    display: grid;
}
```

Rows and columns

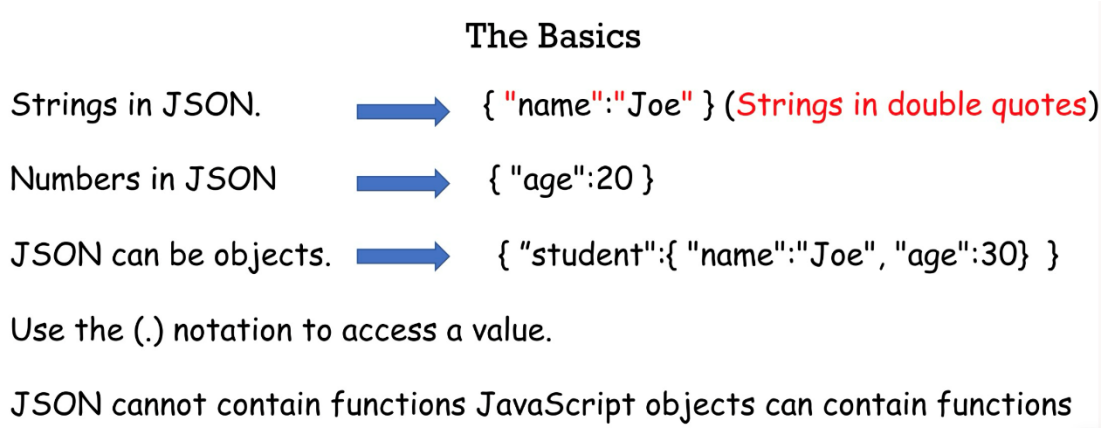
```
grid-row-start: 1; (start at row 1)
grid-row-end: 3;
grid-column-start: 2;
grid-column-end: 3;

grid-row: 1 / 3;
grid-column: 2 / 3;
```

grid-area: 1 / 2 / 3 / 3;

```
body {
      display: grid;
      grid-template-column: 200px 1fr;
}
header { grid-row: span 2; }
footer { grid-row: span 2; }
```

Week 7

JavaScript



JSON.stringify(); **JavaScript** object to JSON object
JSON.parse(); **JSON** object to JavaScript object

array.forEach(function(currentValue, index, arr), thisValue)
function(currentValue, index, arr): This is a function to execute for each element, taking three arguments:
currentValue: The current element being processed in the array.
index (optional): The index of the current element being processed in the array.

arr (optional): The array that forEach() is being applied to.
thisValue (optional): A value to use as this when executing the callback function.

```
let fruits = ['apple', 'banana', 'cherry'];
fruits.forEach(function(item, index) {
      console.log(index, item);
});
```

Output:
0 apple
1 banana
2 cherry

Arrays

```
var name = ``johndoe";

console.log(name.length);   = 7


console.log(name[0]); = 'j'
console.log(name[name.length]); = undefined
console.log(name[name.length-1]); = 'e'
```

Array.length count space

Use of pop() and map():

```
let fruits = ['Apple', 'Banana', 'Cherry'];

// Use pop to remove the last element ('Cherry')
let lastFruit = fruits.pop();

console.log(lastFruit);    // Outputs: Cherry
console.log(fruits);       // Outputs: ['Apple', 'Banana']
```

```
let numbers = [1, 2, 3, 4];
let squares = numbers.map(function(num) {
      return num * num;
});

console.log(squares);   // Outputs: [1, 4, 9, 16]
```

DOM(document edit)

const link = document.querySelector("a"); use “querySelector” to find the reference and store in a variable
link.textContent = "Mozilla Developer Network"; change the content
link.href = "https://developer.mozilla.org"; change the URL

- `Document.getElementById()`, which selects an element with a given `id` attribute value, e.g. `<p id="myId">My paragraph</p>`. The ID is passed to the function as a parameter, i.e. `const elementRef = document.getElementById('myId')`.
- `Document.getElementsByTagName()`, which returns an array-like object containing all the elements on the page of a given type, for example `<p>`s, `<a>`s, etc. The element type is passed to the function as a parameter, i.e. `const elementRefArray = document.getElementsByTagName('p')`.

const sect = document.querySelector("section"); select the “section” part
const para = document.createElement("p"); create a new element with tag “p”
para.textContent = "We hope you enjoyed the ride."; add new <p> element
sect.appendChild(para); append the new <p> element at the end of section
const text = document.createTextNode(
" — the premier source for web development knowledge.",
); create a textnode
const linkPara = document.querySelector("p"); this selector only select first <p> (use *document.querySelectorAll* to select all element)
linkPara.appendChild(text);
sect.removeChild(linkPara); will remove whole line if the *linkPara* have append to some elements
linkPara.remove(); same as above

```
para.style.color = "white";
para.style.backgroundColor = "black";
para.style.padding = "10px";
para.style.width = "250px";
para.style.textAlign = "center";
```

above command use .style to modify the CSS in script

```
<style>
  .highlight {
      color: white;
      background-color: black;
      padding: 10px;
      width: 250px;
      text-align: center;
  }
</style>
para.setAttribute("class", "highlight");
```
same as above 5 lines command

${somevalue} If you omitted the $ and only used {name}, it would not be recognized as an expression to be evaluated, and instead, it would be treated literally as the string "{name}". Thus, the $ is essential for indicating that the enclosed {name} is an expression whose value should be dynamically inserted into the string.

Week 9

wget -p <url> include stylesheets, images and robot.txt

ROBOTS.TXT

Standard for websites to announce crawling preferences.
Simple text file stored in top-level directory of a site.
Specifies which user-agents are permitted to access which resources.

Recursive downloading

wget -r -l N <url>
By default, constrained to only pages on the same domain as that in the webpage URL.
-l N: The level of recursion to permit. (Default if omitted is 5).

wget -m <url>
Uses standard defaults for creating a local copy of a **full** website. (Think about if you really want the entire website).
-w 1: Wait 1 second between each request (to avoid annoying a server).