中国科学院大学
University of Chinese Academy of Sciences

CS101

Network Thinking

Web Programming

zxu@ict.ac.cn
zhangjialin@ict.ac.cn

This can also be the second lecture, after the Network Thinking lecture

Outline

- What is network thinking?
- Network terms
- Connectivity
  - Naming
  - Topology
- Protocol stack
  - The Web over TCP/IP stack
  - Web programming
- Network laws
  - Performance metrics
  - Network effect
- Responsible computing

These slides acknowledge sources for additional data not cited in the textbook

4.2 Basic Web programming

- Introduce basic HTML/CSS/JavaScript
  - Know webpage and **dynamic webpage**
  - Prepare students for the Personal Artifacts project
- Students are suggested to
  - transfer Go programming knowledge to Web programming
  - learn additional knowledge needed
    - by referencing the library of Personal Artifacts examples

Graphics credit:
Siyue Li

Who spent 50% time in creating, and 50% time in coding

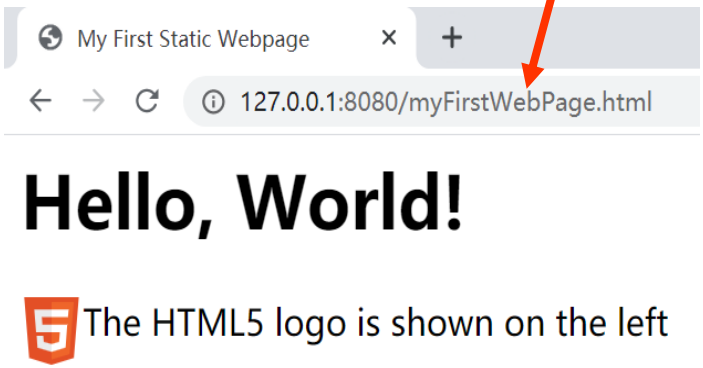
https://cs101.ucas.edu.cn/???

Web client and Web server

- Web client
  - Runs a Web browser
  - Accesses a webpage (Web resource) from a Web server
    - By issuing a Web request with a URL
  - Renders a webpage (Web resource) transferred from a Web server
  - Interacts with the user (e.g., a human being)
- Web server
  - Hosts Web pages (Web resources)
  - Accepts and process Web requests
  - and sends back responses

Assume Web client and Web server are both a student's laptop computer

- A Web browser uses **URL** to access a **Webpage**
  - Also **renders** a webpage and interact with the user
- Web server
  - Hosts Web pages (Web resources)
  - Accepts and process Web requests and sends back responses



Browser display
at the Web client

```
> cat myFirstWebPage.html
<html>
   <head>
      <meta charset="utf-8">
      <title>My First Static Webpage</title>
   </head>
   <body>
      <h1> Hello, World! </h1>
      <p>
          <img
src="https://www.w3.org/html/logo/downloads/HTML5_Badge.svg"
          style="float:left;width:30px;height:30px;">
          The HTML5 logo is shown on the left
      </p>
      <script>
      </script>
   </body>
</html>
>
```

A webpage file in the Web server

Write your first Web program

- Start a Web server

```
> cat WebServer.go
package main
import "net/http"
func main() {
   http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
            http.ServeFile(w, r, r.URL.Path[1:])
   })
   http.ListenAndServe(":8080", nil)
}
> go build WebServer.go
> ./WebServer &
> [1] 442
>
>
> kill 442
```

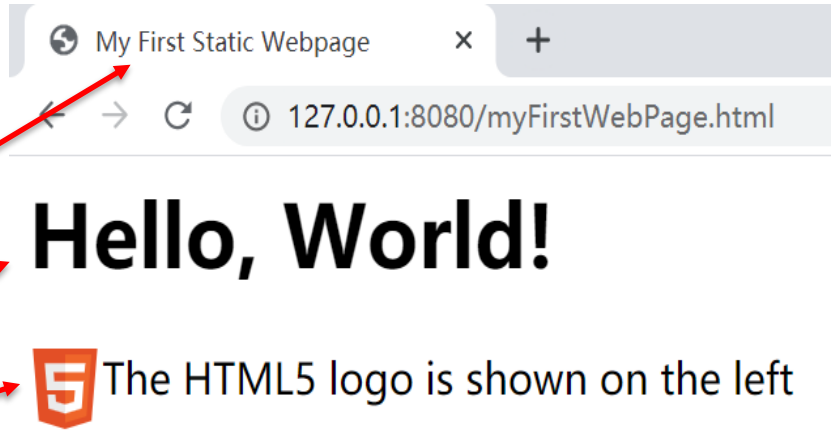
Run program WebServer in background

WebServer is ready to accept HTTP requests (442 is WebServer process ID)

Stop WebServer

Write your first Web program

- Enter “http://127.0.0.1:8080/myFirstWebPage.html” in a Web browser
  - 127.0.0.1 is localhost

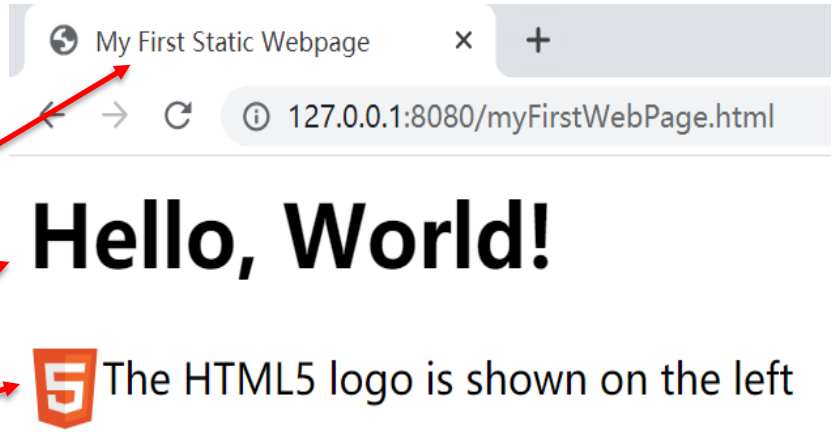



```
> cat myFirstWebPage.html
<html>
  <head>
    <meta charset="utf-8">
    <title>My First Static Webpage</title>
  </head>
  <body>
    <h1> Hello, World! </h1>
    <p>
      <img src="https://www.w3.org/html/logo/downloads/HTML5_Badge.svg"
      style="float:left;width:30px;height:30px;">
      The HTML5 logo is shown on the left
    </p>
    <script>
    </script>
  </body>
</html>
>
```

HTML

Write your first Web program

- Enter “http://127.0.0.1:8080/myFirstWebPage.html” in a Web browser

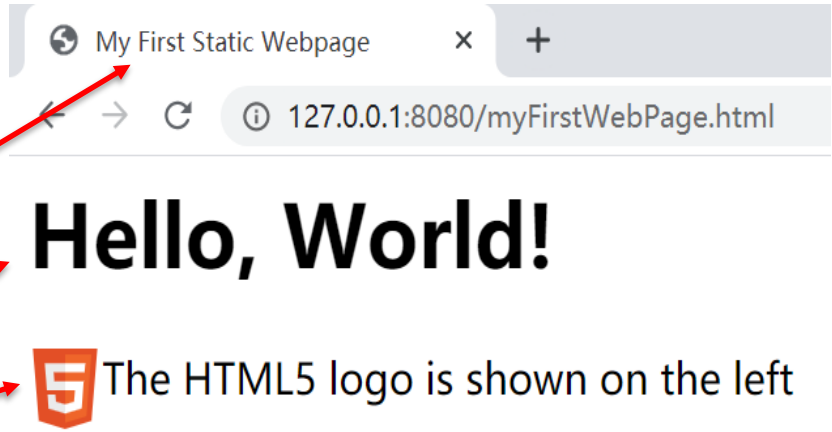



```
> cat myFirstWebPage.html
<html>
  <head>
    <meta charset="utf-8">
    <title>My First Static Webpage</title>
  </head>
  <body>
    <h1> Hello, World! </h1>
    <p>
      <img src="https://www.w3.org/html/logo/downloads/HTML5_Badge.svg"
      style="float:left;width:30px;height:30px;">
      The HTML5 logo is shown on the left
    </p>
    <script>
    </script>
  </body>
</html>
>
```

CSS

Write your first Web program

- Enter “http://127.0.0.1:8080/myFirstWebPage.html” in a Web browser

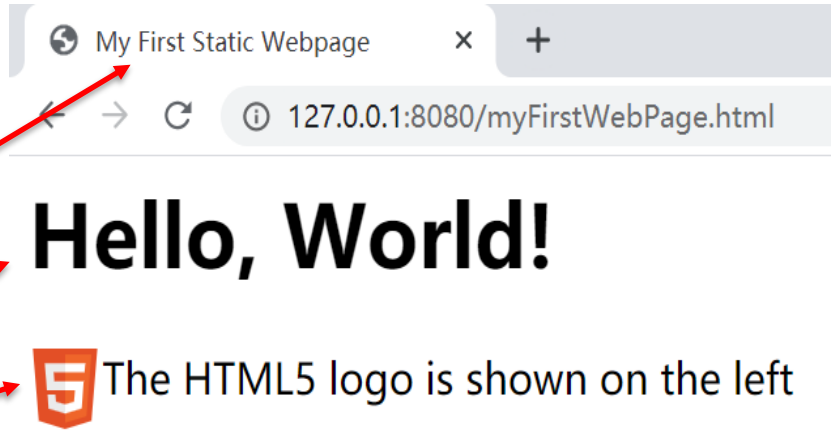



```
> cat myFirstWebPage.html
<html>
  <head>
    <meta charset="utf-8">
    <title>My First Static Webpage</title>
  </head>
  <body>
    <h1> Hello, World! </h1>
    <p>
      <img src="https://www.w3.org/html/logo/downloads/HTML5_Badge.svg"
      style="float:left;width:30px;height:30px;">
      The HTML5 logo is shown on the left
    </p>
    <script>
    </script>
  </body>
</html>
>
```

No JavaScript code in this webpage
This is a static webpage

Write your first Web program

- Enter “http://127.0.0.1:8080/myFirstWebPage.html” in a Web browser




```
> cat myFirstWebPage.html
<html>
  <head>
    <meta charset="utf-8">
    <title>My First Static Webpage</title>
  </head>
  <body>
    <h1> Hello, World! </h1>
    <p>
      <img src="https://www.w3.org/html/logo/downloads/HTML5_Badge.svg"
      style="float:left;width:30px;height:30px;">
      The HTML5 logo is shown on the left
    </p>
    <script>
    </script>
  </body>
</html>
>
```

- List things common to all elements in <head>…</head>
- Sequentially list elements to be displayed in <body>…</body>
  This example has 3 elements: **h1, p, img**
- CSS to specify element style
- JavaScript code to operate on elements

staticChildrensDay.html: code and output

- Now we use JavaScript

Render the **Content** according to **Style** at the place indicated by **Element ID**

"2021.06.01";
~~60px, purple~~
"childrensDay"

```
> cat staticChildrensDay.html
<html>
   <head>
      <meta charset="utf-8">
      <title>Display the date of next Children's Day</title>
   </head>
   <body>
      <h1 style="text-align: center">Date of Next Children's Day</h1>
      <p style="text-align: center" id="childrensDay" ></p>
      <script>
         var x = document.getElementById("childrensDay");
         x.style.fontSize = "60px";
         x.style.color = "purple";
         x.innerHTML = "2021.06.01";
      </script>
   </body>
</html>
>
```

Element ID

Style

Content

staticChildrensDay.html: code and output

- Now we use JavaScript

Render the **Content** "2021.06.01";
according to **Style** 60px, purple
at the paragraph indicated "childrensDay"
by **Element ID**

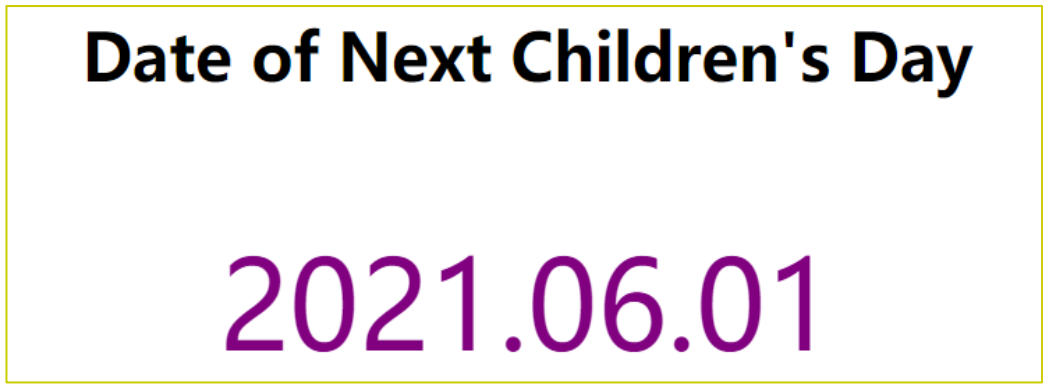
```
> cat staticChildrensDay.html
<html>
   <head>
      <meta charset="utf-8">
      <title>Display the date of next Children's Day</title>
   </head>
   <body>
      <h1 style="text-align: center">Date of Next Children's Day</h1>
      <p style="text-align: center" id="childrensDay" ></p>
      <script>
         var x = document.getElementById("childrensDay");
         x.style.fontSize = "60px";
         x.style.color = "purple";
         x.innerHTML = "2021.06.01";
      </script>
   </body>
</html>
>
```

Element ID

Style

Content

Display Content
with 60 pixels font size
and purple color



ChildrensDay.html: code and output

- Now use JavaScript to create a dynamic webpage

```
> cat staticChildrensDay.html
<html>
   …
   <body>
      <h1 style="text-align: center">Date of Next Children's Day</h1>
      <p style="text-align: center" id="childrensDay" ></p>
      <script>
         var x = document.getElementById("childrensDay");
         x.style.fontSize = "60px";
         x.style.color = "purple";
         var date = new Date;
         var year = date.getFullYear();
         var month = date.getMonth() + 1;
         if (month >= 6) year = year + 1;
         x.innerHTML = "June 1, " + year;
      </script>
   </body>
</html>
>
```

Element ID
Style
Content

ChildrensDay.html: code and output

- How many elements in this webpage?
  - 2. The **h1** (level-1 header) and the **p** (paragraph) elements

```
> cat staticChildrensDay.html
<html>
   …
   <body>
      <h1 style="text-align: center">Date of Next Children's Day</h1>
      <p style="text-align: center" id="childrensDay" ></p>
      <script>
         var x = document.getElementById("childrensDay");
         x.style.fontSize = "60px";
         x.style.color = "purple";
         var date = new Date;
         var year = date.getFullYear();
         var month = date.getMonth() + 1;
         if (month >= 6) year = year + 1;
         x.innerHTML = "June 1, " + year;
      </script>
   </body>
</html>
>
```

Element ID
Style
Content

Date of Next Children's Day

June 1, 2021

JavaScript is an object-oriented language

- An object puts data type and functions together
  - Data structure + methods to operate the data structure
  - Use the dot notation to access methods of an object
    - date.getMonth: use the getMonth method of the date object
    - document.getElementById: use the getElementById method of the document object

```
<html>
<body>
<p id="myDate"></p>
<p id="myYear"></p>
<p id="myMonth"></p>
<script>
var date = new Date();
var year = date.getFullYear();
var month = date.getMonth() + 1;
document.getElementById("myDate").innerHTML = date;
document.getElementById("myYear").innerHTML = year;
document.getElementById("myMonth").innerHTML = month;
</script>
</body>
</html>
```

This Web code displays three paragraphs:

- Full information of the current date
- Current year
- Current month

Fri Apr 16 2021 19:19:50 GMT+0800 (中国标准时间)
2021
4

Notes:

- Date is a system provided object
- document is a system provided object
- var date = new Date(); create a new object
- Read textbook to see why month needs to add 1

The Web code can be rewritten as follows

- Note that the line of code
  - document.getElementById("myDate").innerHTML = date;
- Is broken down into two shorter lines of code
  - var x = document.getElementById("myDate");
  - x.innerHTML = date;

Original Code

```
<html>
<body>
<p id="myDate"></p>
<p id="myYear"></p>
<p id="myMonth"></p>
<script>
var date = new Date();
var year = date.getFullYear();
var month = date.getMonth() + 1;
document.getElementById("myDate").innerHTML = date;
document.getElementById("myYear").innerHTML = year;
document.getElementById("myMonth").innerHTML = month;
</script>
</body>
</html>
```

New Code

```
<html>
<body>
<p id="myDate"></p>
<p id="myYear"></p>
<p id="myMonth"></p>
<script>
var date = new Date();
var year = date.getFullYear();
var month = date.getMonth() + 1;
var x = document.getElementById("myDate");
x.innerHTML = date;
var y = document.getElementById("myYear");
y.innerHTML = year;
var z = document.getElementById("myMonth");
z.innerHTML = month;
</script>
</body>
</html>
```