网络思维-1

网络思维概述，名词术语，Web编程

zxu@ict.ac.cn

提纲　　Acu-Exams-CP

- 什么是网络思维
- 网络名词术语
- Web编程
  - 网页执行流程
  - 网页结构（DOM）

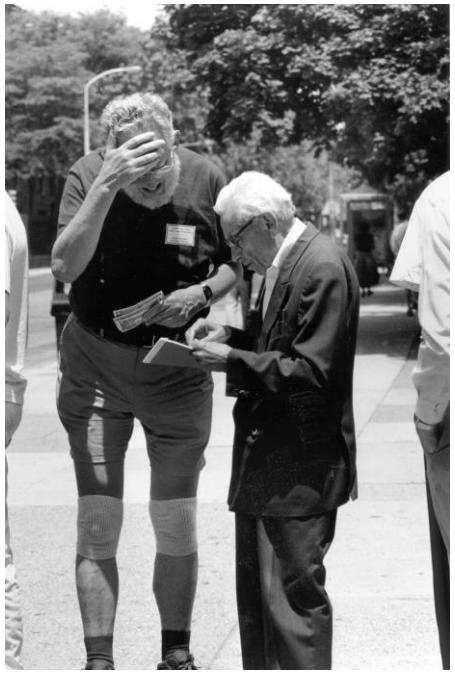
课件中包含教科书未包括的素材引用，特此致谢

1. 什么是网络思维？

- 通过连通性与协议栈，研究多个节点连接而成的网络
  - 很多问题涉及用户/数据/算法/部件的连接体，而非单体
- 连接体就是网络，即多个节点连接或通信的整体
  - 必有连接，可有通信
- 网络是客体（object，宾语）：文献网
  - 全球计算机科学文献网络，连接是引用；无通信
- 网络是主体（subject，主语）：机群（cluster of computers）
  - 机群 算出 文献网；机群的节点之间有通信
- 三类网络：硬件互联（interconnect）、社会网络、互联网
  - 以互联网为主要场景

网络思维使人们发现了很多有趣现象

- E.g., what is your Erdös Number?
  - Measuring interdisciplinary nature of modern research
    - https://mathscinet.ams.org/mathscinet/freeTools.html?version=2
- Paul Erdös（爱尔迪西）, 匈牙利数学家 (1913–1996)
- “Master of Collaboration”, 1400篇论文，500合著者
- Erdös Number = 0 → Erdös himself
- Erdös Number = 1 → Erdös’ coauthor
- Erdös Number = 2 → Erdös’ coauthor’s coauthor
- Erdös Number = 3 → Erdös’ coauthor’s coauthor’s coauthor



金芳蓉教授网站
https://mathweb.ucsd.edu/~fan/photo/ep.html

网络思维使人们发现了很多有趣现象

- E.g., what is your Erdös Number?
  - Measuring interdisciplinary nature of modern research
    - https://mathscinet.ams.org/mathscinet/freeTools.html?version=2

- Paul Erdös, 匈牙利数学家 (1913–1996)
- “Master of Collaboration”, 1400篇论文，500合著者
- Erdös Number = 0 → Erdös himself
- Erdös Number = 1 → Erdös’ coauthor
- Erdös Number = 2 → Erdös’ coauthor’s coauthor
- Erdös Number = 3 → Erdös’ coauthor’s coauthor’s coauthor

- 徐志伟：Erdös Number = ？　　计算机系统
- 张家琳：Erdös Number = ？　　计算机理论

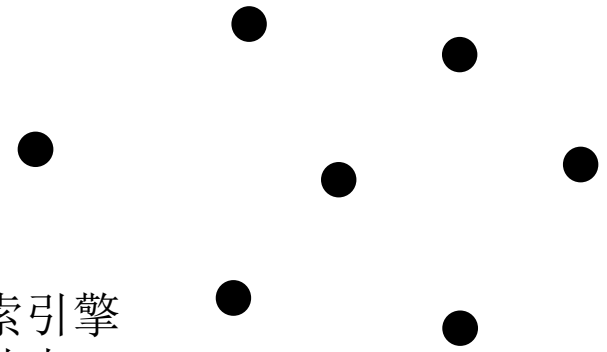
网络思维使人们发现了很多有趣现象

- E.g., what is your Erdös Number?
  - Measuring interdisciplinary nature of modern research
    - https://mathscinet.ams.org/mathscinet/freeTools.html?version=2
- Paul Erdös, 匈牙利数学家 (1913–1996)
- “Master of Collaboration”, 1400篇论文，500合著者
- Erdös Number = 0 → Erdös himself
- Erdös Number = 1 → Erdös’ coauthor
- Erdös Number = 2 → Erdös’ coauthor’s coauthor
- Erdös Number = 3 → Erdös’ coauthor’s coauthor’s coauthor

- 徐志伟：Erdös Number = ？ 计算机系统
- 张家琳：Erdös Number = ？ 计算机理论
- Zhiwei Xu: 4
- Jialin Zhang: 3

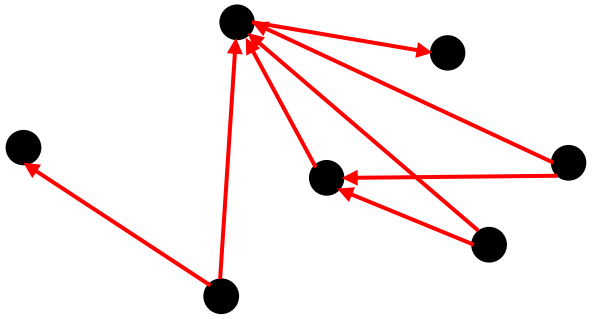
网络思维催生新概念、新方法

- 第一代 vs. 第二代搜索引擎
- 1st generation search engines
  - Computed search results by matching the keywords in search queries to the contents of webpages (*nodes*)
  - Only utilized nodes of the network of webpages
- 2nd generation search engines
  - Around 1996, Jon Kleinberg, Robin Li (李彦宏), and Larry Page observed a phenomenon:
    - Web links also significantly influence the relevance of search results
  - Utilized both nodes and interconnections to develop the 2G search engines with better results
    - More fully utilizes network thinking and created Google and Baidu, serving billions of users and generating annual revenue over $100 billion



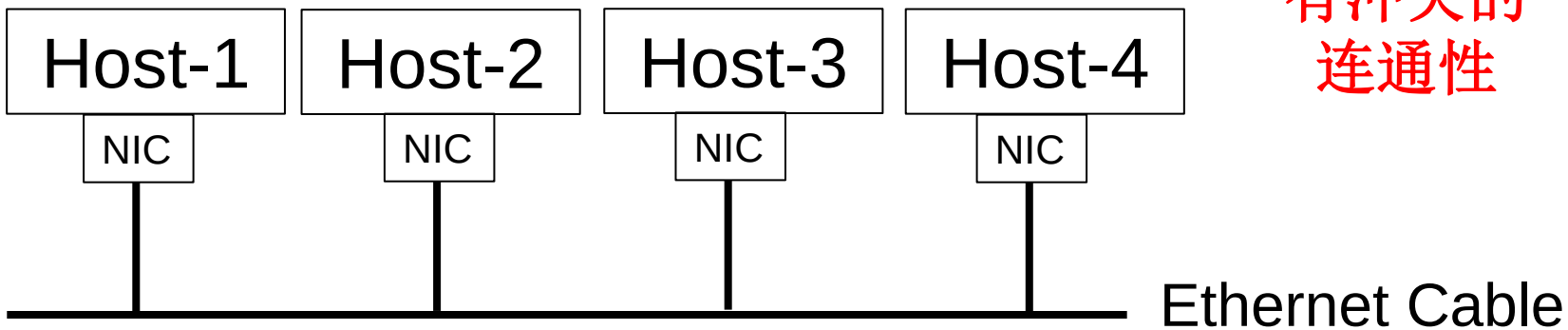
第一代搜索引擎
只利用了节点

第二代搜索引擎
利用了节点和边



化解冲突的巧妙方法

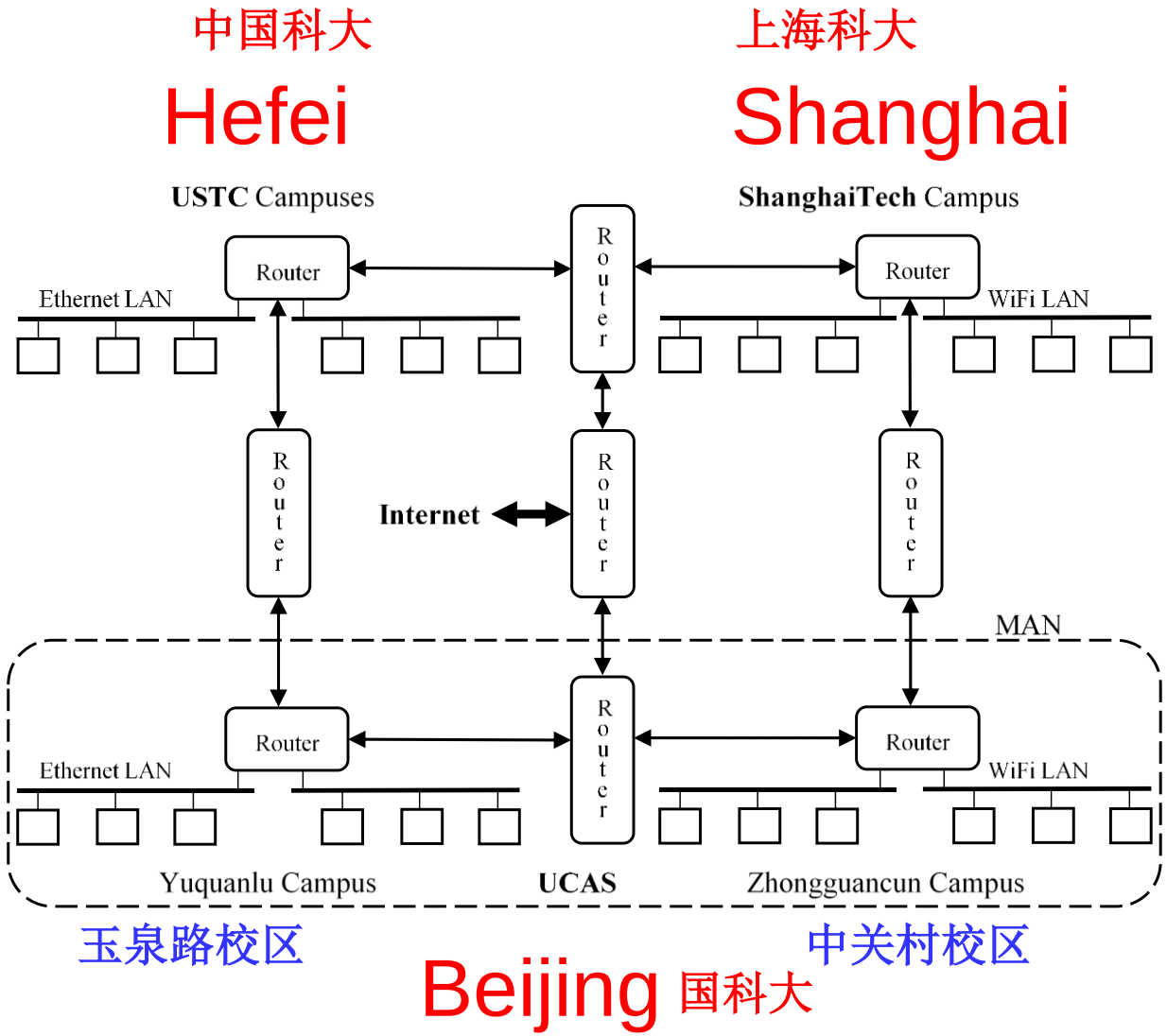
- Four hosts connected by an Ethernet
  - NIC (network interfacing circuitry) for networking operations
- Conflict example: Host-3与4在通信时，Host-1与2试图通信
  - Host-1 tries to send a message to Host-2, while the cable is used by Host-3 communicating with Host-4
- Exponential backoff to resolve conflict　指数退避方法
  - When first try fails, Host-1 waits for a random time in [0, T]
  - When second try fails, Host-1 waits for a random time in [0, 2T]
  - When third try fails, Host-1 waits for a random time in [0, 4T]



有冲突的连通性

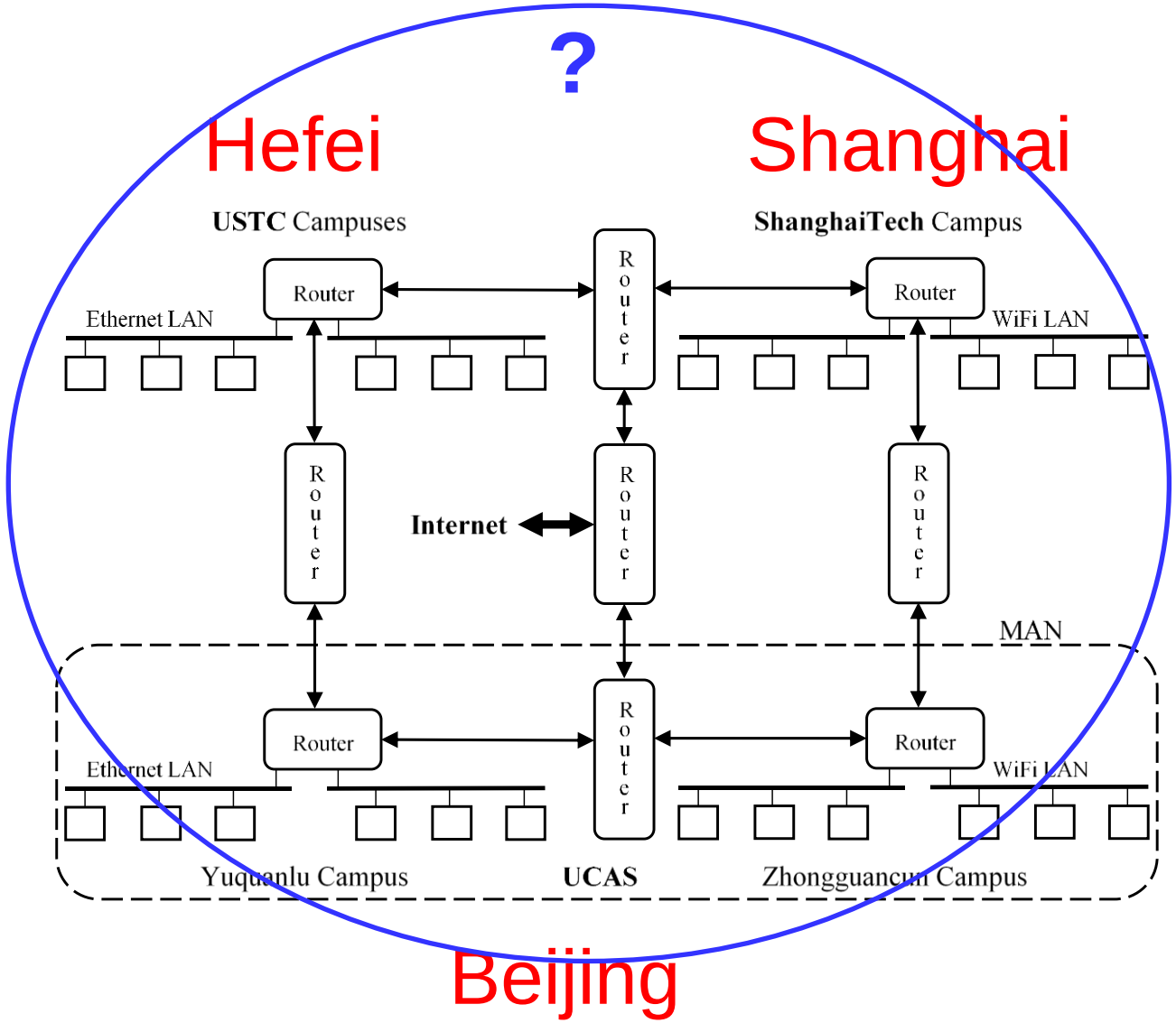
2. 网络名词术语

- LAN 局域网
  Local Area Network
- MAN 城域网
  Metropolitan Area Network
- WAN 广域网
  Wide Area Network



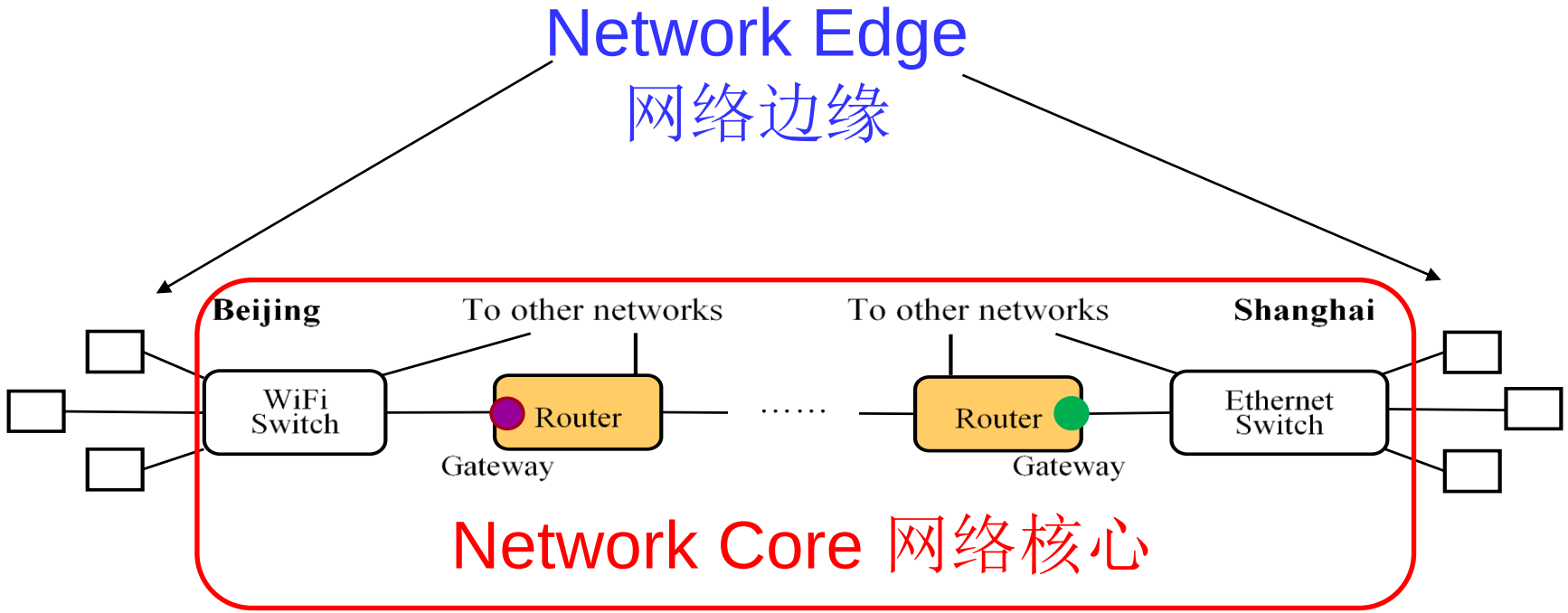
Network terms

- **ISP** Internet Service Provider
  互联网服务提供商
  - An institution providing Internet connection services
  - CSTNET for …ac.cn
  - CERNET for …edu.cn



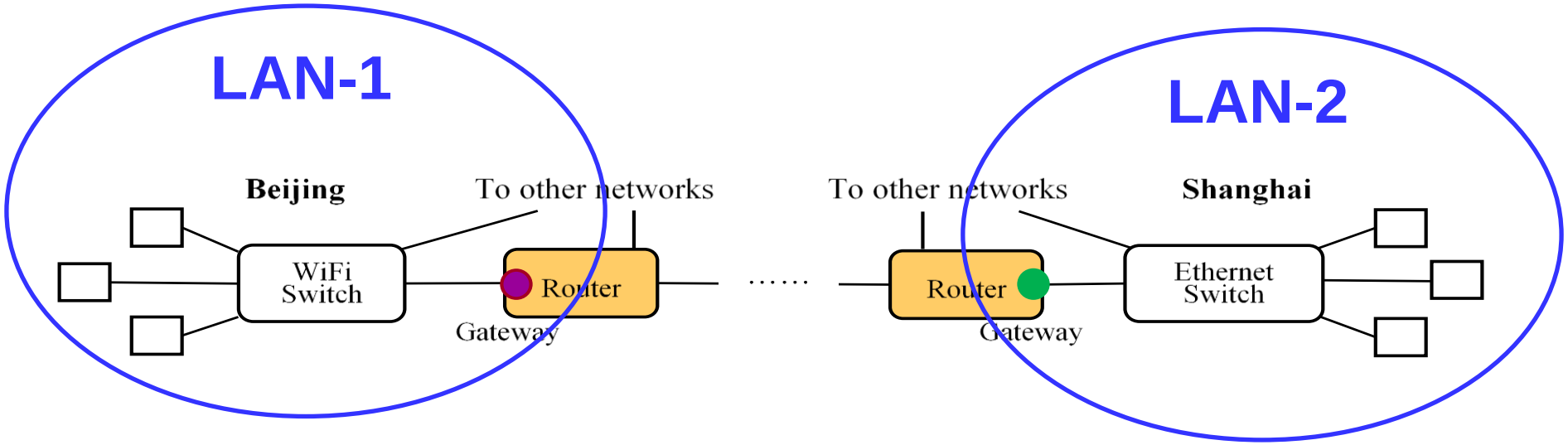
两类网络节点
宿主节点（宿主机）、组网设备节点

- Hosts: client nodes and server nodes at the network edge
  - 6 edge devices (hosts) are shown　　宿主机构成网络边缘
  - Client devices: laptop and desktop computers, smartphones, etc.　　客户端设备
  - Server devices: servers, Internet datacenters, supercomputers, etc.　　服务器设备
- Networking devices in the core of the network
  - 4 networking devices are shown　　组网设备构成网络核心



Internal network and outside network of an organization

- 网关：连接局域网与外界的路由器（端口）
- Gateway of network LAN-1 (purple)
  - The router connecting a network to the outside
    - In more detail, the port address of the router
- Gateway of network LAN-2 (green)



3. 网页编程

- HTML/CSS/JavaScript入门知识，提升学习能力
- 学习方法建议
  - 将已学到的Go编程知识拓展到Web编程
    - 注意语法不同点，例如每条语句后要加“;”
  - 通过例子学习新知识；实验课有详解，与助教一起做一遍
  - 提升自己的学习能力与创造性表达
    - 可参考往届同学个人作品库 https://course.things.ac.cn:10088/exp/personal_artifact

Graphics credit:
Siyue Li 李思悦 2019级
50%时间创造
50%时间开发

万维网网址（URL）入门知识

- Uniform Resource Locator

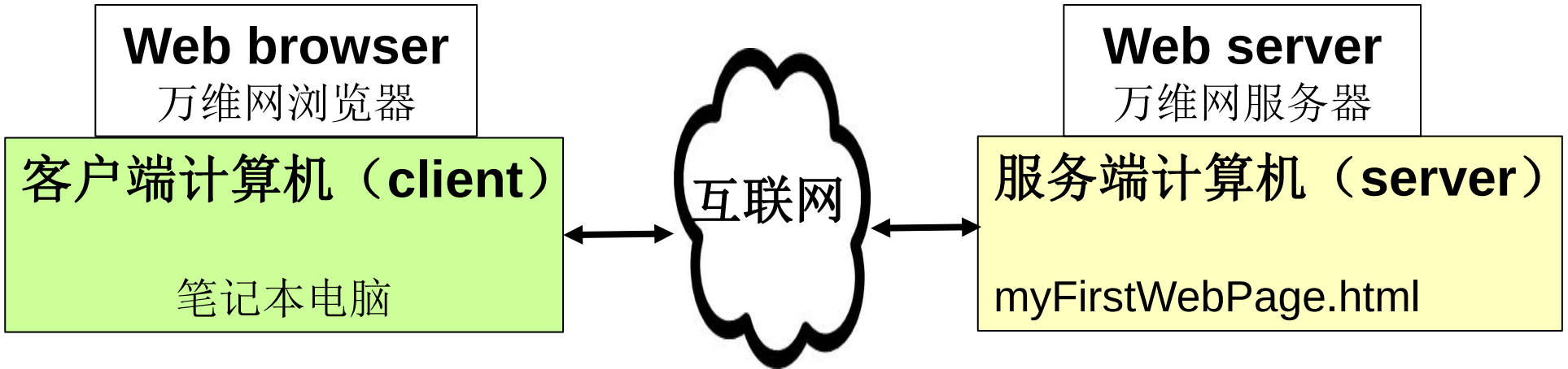
http :// cs101.ucas.edu.cn /中文/

协议 网站（域名或IP地址） 路径

- 其他协议：
  - file：访问本计算机的文件
  - ftp：访问互联网上任意计算机的文件
  - https：安全地访问Web资源
  - mailto：访问电子邮件地址

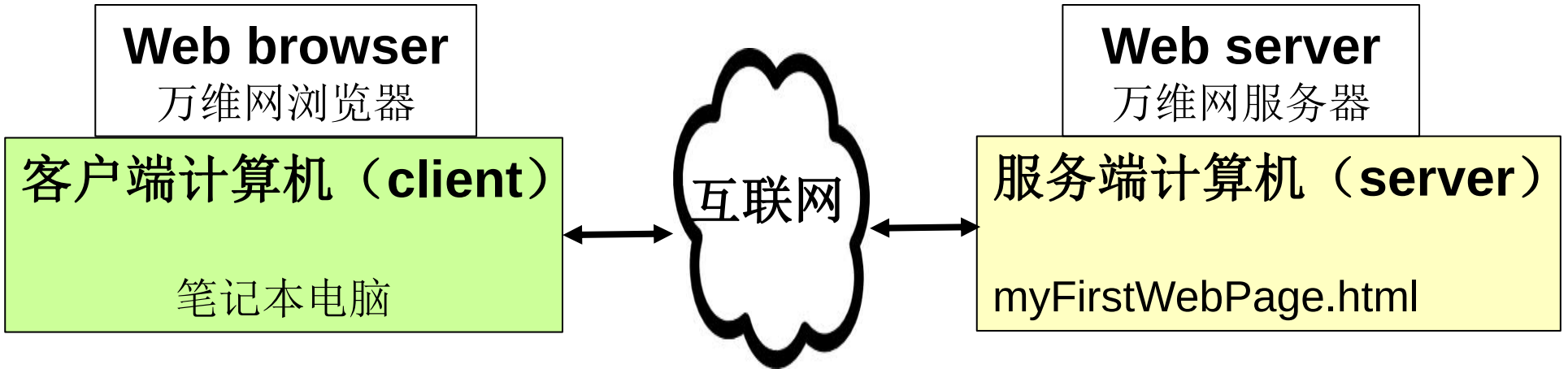
什么是Web客户端（客户机），什么是Web服务器？

- 客户机上运行浏览器，使用网址从Web服务器获取网页文件到客户机，再由浏览器处理并展示网页。（处理并展示=render）
  - 此时，Web服务器已经在运行



什么是Web客户端（客户机），什么是Web服务器？

- 客户机上运行浏览器，使用网址从Web服务器获取网页文件到客户机，再由浏览器处理并展示网页。
  - 此时，Web服务器已经在运行
- 浏览器：　　browser
- 万维网服务器：　　Web server
- 网址：　　URL
- 网页：　　Webpage
- 网页文件：　　myFirstWebPage.html



开发你的第一个网页（静态网页）演示

- 在你的云环境中执行WebServer.go

```
> cat WebServer.go
package main                    // WebServer.go，不支持HTTPS
import “net/http“
func main() {
   http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
            http.ServeFile(w, r, r.URL.Path[1:])
   })
   http.ListenAndServe(":8081", nil)
}
> go build WebServer.go
> ./WebServer &
> [1] 69474
>
…
>
> kill 69474
```

Run program WebServer in background
后台运行WebServer程序

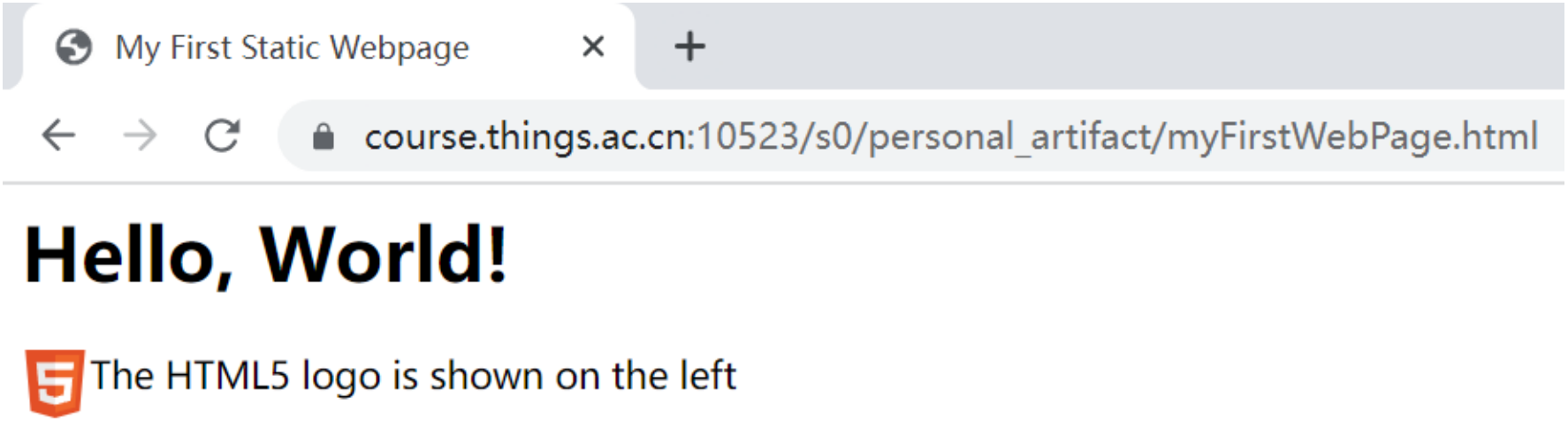
WebServer 就绪
(69474 is WebServer process ID)　进程号、进程ID

Stop WebServer
终止ID为69474的进程

你的第一个网页（html文档）及其网页显示

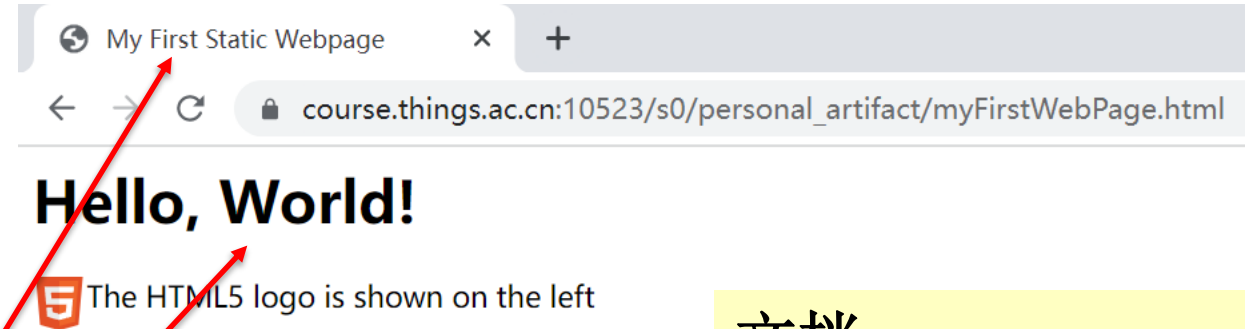
- 在浏览器网址栏输入网址　　这个网址每年可能不一样，今年是：
  https://course.things.ac.cn:10179/s0/personal_artifact/myFirstWebPage.html
- 从而得到网页显示

- 客户机上运行浏览器，使用网址从Web服务器获取网页文件到客户机，再由浏览器处理并展示网页。（处理并展示=render）



你的第一个网页（html文档）及其网页显示

- 网页编程采用文档对象模型（**DOM**，Document Object Model）
  - 网页 = 文档
  - 浏览器展示全部网页
    - 而不是像Go程序那样
逐条语句串行执行



网页头

网页体

```
> 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>          此处JavaScript代码为空
    </script>
  </body>
</html>
>
```

文档

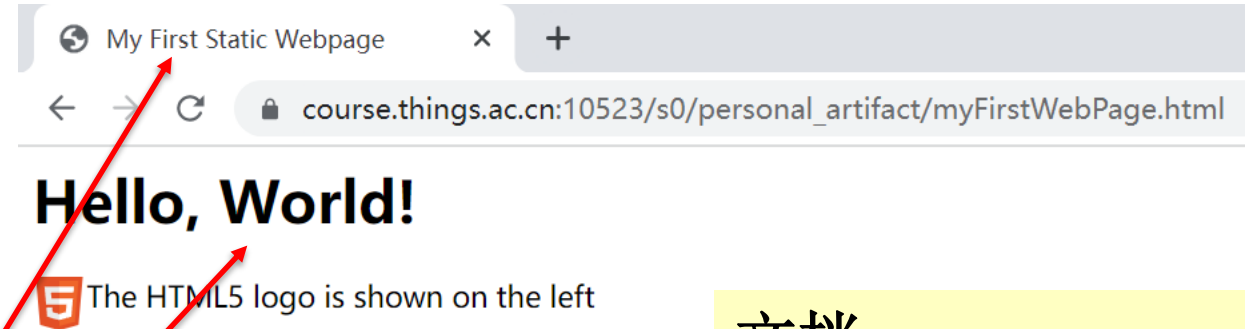
- 包括“头”和“体”
- 由若干元素组成
- 元素被标签括起来
- 头部说明网页全局

HTML网页内容

被<html> … </html>标签括起来

你的第一个网页（html文档）及其网页显示

- 网页编程采用文档对象模型（**DOM**，Document Object Model）
  - 网页 = 文档
  - 浏览器展示全部网页
    - 而不是像Go程序那样
      逐条语句串行执行



网页头

网页体

```
> 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>          此处JavaScript代码为空
    </script>
  </body>
</html>
>
```

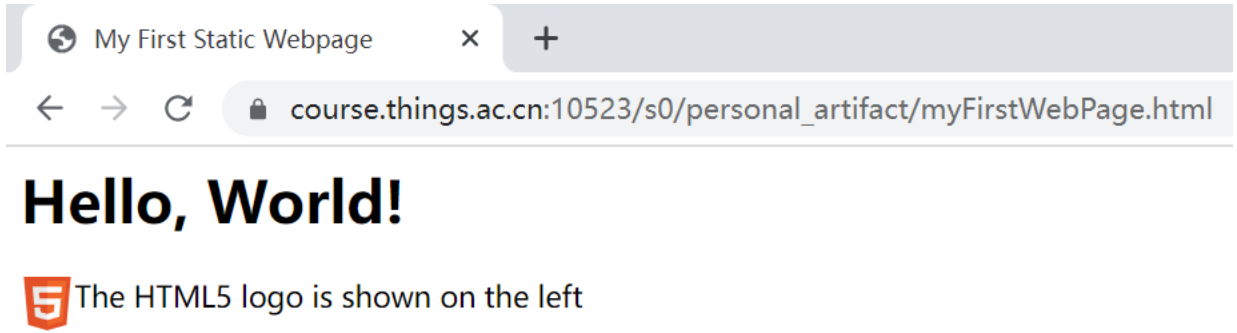
文档

- 包括“头”和“体”
- 由若干元素组成
- 元素被标签括起来
- 头部说明网页全局

HTML网页内容

被<html> … </html>标签括起来

你的第一个网页（html文档）及其网页显示

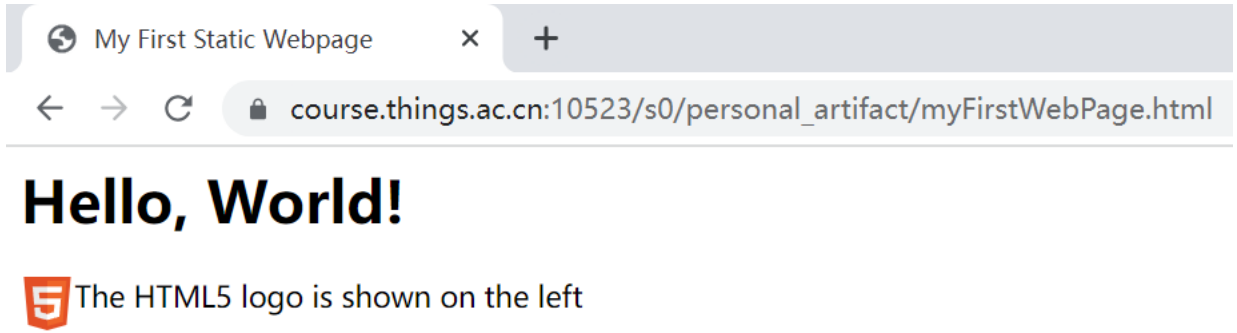



```
> cat myFirstWebPage.html
<html>
    <head>
        <meta charset="utf-8">
        <title>My First Static Webpage</title>
    </head>
    <body>
        <h1> Hello, World! </h1>
        <p>
            <img src="https://www.w3.org/html/logo/downloads/HTML5_Badge.svg"
            style="float:left;width:30px;height:30px;">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

CSS 展示风格

图像浮在文字左边，30像素宽，30像素高

你的第一个网页（html文档）及其网页显示



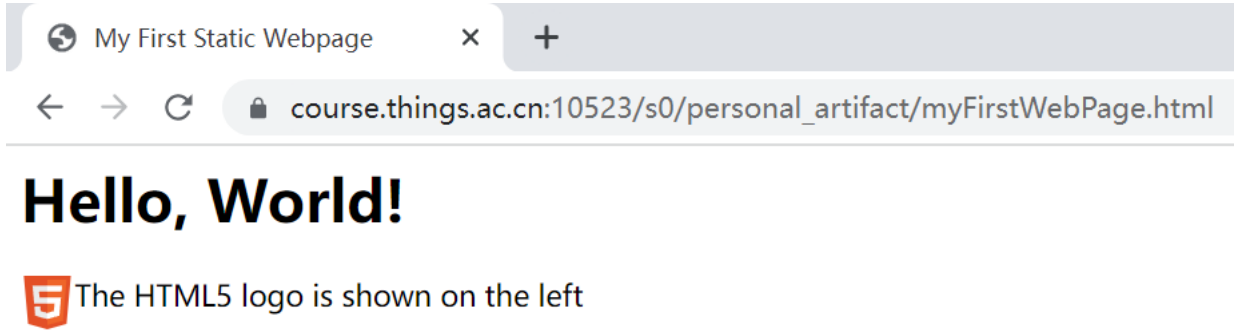

```
> cat myFirstWebPage.html
<html>
    <head>
        <meta charset="utf-8">
        <title>My First Static Webpage</title>
    </head>
    <body>
        <h1> Hello, World! </h1>
        <p>
            <img src="https://www.w3.org/html/logo/downloads/HTML5_Badge.svg"
            style="float:left;width:30px;height:30px;">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

No JavaScript code in this webpage

This is a static webpage

静态网页可不用**JavaScript**

你的第一个网页（html文档）及其网页显示




```
> 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>
>
```

Web编程要点

- 在head里列出所有元素共性的东西
- 在body里列出网页(顺序)展示的**元素**
  - **此例子共有三个元素；p元素包含img元素**
  - 顺序是空间顺序
- CSS指明元素的显示风格
- JavaScript代码操作元素（动态）

staticChildrensDay.html: code and output

- 使用JavaScript的例子

- Render the **Content** "2021.06.01";
- according to **Style** ~~60px, purple~~
- at the place indicated by **Element ID** "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 元素ID

Style 风格

Content 内容

- 定义段落元素的ID = childrensDay
- 使用document.getElementById函数，使得变量x指称该段落元素
- x.style指称该段落元素的风格
- x.innerHTML指称该段落元素的内容

staticChildrensDay.html: code and output

- 加上两条风格说明

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

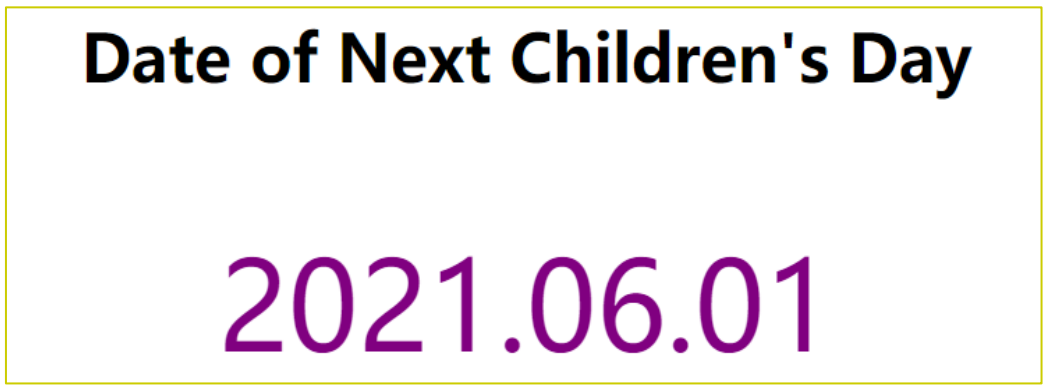
```
> cat staticChildrensDay.html
<html>
   <head>
      <meta charset="utf-8">
      <title>Display the date of next Children's Day</title>
   </head>
   <body>
      <h1 style="text-align: center">Date of Next Children's Day</h1>
      <p style="text-align: center" id="childrensDay" ></p>
      <script>
         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

Display Content
with 60 pixels font size
and purple color



ChildrensDay.html: code and output

- 使用JavaScript 产生动态网页
  - 显示的日期随时间正确变化
    - 本网页有多少元素？2个。第二个元素是paragraph <p…></p>

```
> cat ChildrensDay.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
JS是面向对象的语言

- 对象：将数据结构和操作数据的方法放在一个抽象中
  - Object: Data structure + methods to operate the data structure
  - 使用点号记号调用某一对象的某一方法
    - Use the dot notation to access methods of an object
    - date.getMonth: call the getMonth method of the date object
    - document.getElementById: call the getElementById method of the document object

系统提供很多built-in对象，如
Date
Document
只需了解极少几个

```
<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>
```

var date = new Date;
产生新对象并赋值（当前时刻）

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>
```