1. HTTP –
   1. 应用层
   2. 客户端-服务端技术的基础
   3. 可以用于充当不同协议之间的桥梁
   4. 操作的基础对象是URL资源
   5. 消息交换时通过 请求-响应
   6. Stateless协议（一个请求，一次链接）。为了实现对话使用cookies

Структура запросов:

HTTP请求的结构

1. 起始
   1. URI（URL），请求方法 [HTTP版本]（国内这两个东西被分开，所以有四层）
      1. HTTP方法
         1. OPTIONS – 定义服务器的可能。获取服务器所支持的方法
         2. GET – 请求指定页面信息并返回主体（打开网站）
         3. HEAD ­– 同GET，但是在回应中没有主体
         4. POST – 转交资源。向指定资源提交数据进行处理。
         5. PUT – 将请求包含的内容上传到所示URI。从客户端向服务器传送指定数据取代文档的内同
         6. DELET – 请求服务器删除指定页面
   2. 标题
      1. HOST: 客户端信息（主机名，端口号）
      2. USER-AGENT：识别客户使用的游览器，操作系统等等
      3. ACCEPT：告知服务器客户端可以处理的内容格式
      4. CONNECTION：结束后断开链接还是保持等待下一个请求
   3. 消息主体

Структура ответов HTTP：

HTTP回应的结构

1. 起始：状态行由协议版本、数字形式的状态代码、及相应的状态描述，各元素之间以空格分隔。

状态码（код состояния）

由三位数组成，第一位数时状态分类：

* 1. 《1》 – 信息
  2. 《2》 – 成功
  3. 《3》 – 重定向
  4. 《4》 – 客户端错误
  5. 《5》 – 服务端错误

后两位没有分类作用

1. 标题
   * 1. Server：服务器名字
     2. Last-Modified(Time)：文档最后一次编辑时间
     3. Content-Type：表示后面的文档属于什么类型
     4. Content-Language
2. 消息主体（响应正文）

HTTP请求方法

* + - 1. OPTIONS – 定义服务器的可能。获取服务器所支持的方法
      2. GET – 请求指定页面信息并返回主体（打开网站）
      3. HEAD ­– 同GET，但是在回应中没有主体
      4. POST – 转交资源。向指定资源提交数据进行处理。
      5. PUT – 将请求包含的内容上传到所示URI。从客户端向服务器传送指定数据取代文档的内容
      6. DELET – 请求服务器删除指定页面

1. HTML标记语言

特征：

* 1. 互联网的基础标记语言
  2. 用游览器进行解析并且被标示为文件形式
  3. 是SGML（标准通用标记语言）的子集

结构：

* + - * 1. 由元素组成
        2. 元素的开头以及结尾由тег标签标记
        3. 由四部分组成。第一部分是声明位HTML5文档的开头；第二部分是声明HTML完整页面的html元素；第三部分是头部元素head，其中包含了文档的元数据；第四部分是可见元素body
        4. 元素可以为空，只有一个开头标签，且可以拥有属性。
        5. 允许嵌套

1. DOM对象模型
   1. 平台无关接口，允许程序以及脚本对HTML文件的内容进行读取
   2. 文档在DOM中为树形
   3. 各个节点的关系为父子关系
   4. 用于动态地改变HTML文档
2. 关于form方法

详见[链接](https://www.runoob.com/tags/tag-form.html)

1. CSS
   1. 定义样式的法则
   2. 格式包含选择器以及定义块组成

选择器，选择器{

特征：值；

特征：值：

}

* 1. 当对一个元素附上多种样式时，会根据优先级进行选择：
     1. 用！important创建的特征
     2. 在标签中创建的特征
     3. 在选择器中包含#id的特征
     4. 根据选择器中.class以及:pseudoclass的数目
     5. 根据选择器中标签名字的数目
  2. 导入css的方式
     1. 在head中写出style标签，属性标为type =”text / css”
     2. 外部写一个.css文件，然后在head中导入。写一个 <link rel="stylesheet" type="text/css" href="mystyle.css">
     3. 在body的属性中写一个style属性<p style="font-family:arial;color:red;font-size:20px;"></p>
  3. 选择器：说白了就是作用标签的名字。可以使用特殊的通配符以及属性来进行选择

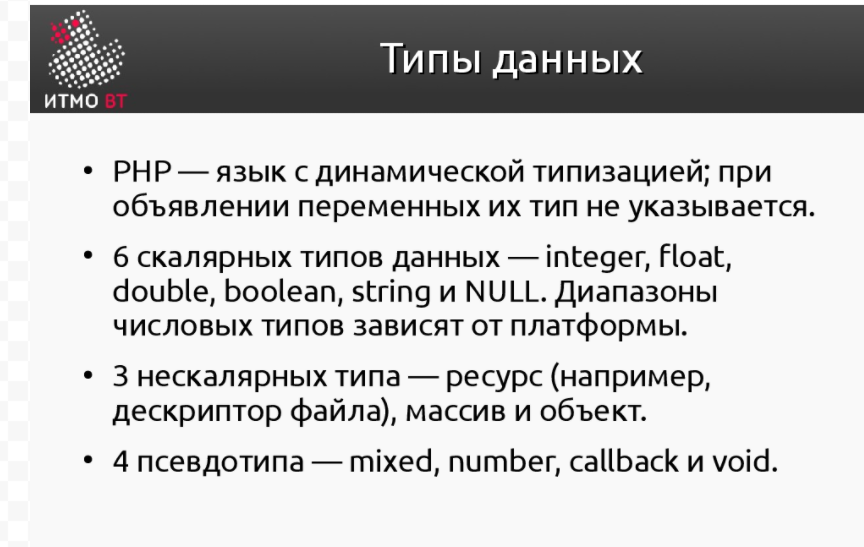
1. Sass,LESS,SCSS
   1. 首先声明，Sass和SCSS实际上一个东西。SCSS是新引进的Sass语法。区别为两个文件后缀名一个是.sass，一个是.scss。同是Sass依靠缩进进行代码的分句，没有大括号和分号，SCSS则与CSS相似
   2. 同：
      1. 都是动态样式语言
      2. 函数，变量，继承，运算等等
      3. 作用域-局部修改
      4. 在css中用javascript表达式赋值
   3. 异：
      1. Less基于javascript，在客户端处理。Sass基于Ruby，在服务端处理
      2. 关于变量在Less和Sass中的唯一区别就是Less用@，Sass用$。
   4. 兼容性：游览器可能会不兼容Less或者Sass，需要特别的转换器将他们转换成一般的CSS
2. 客户端的JavaScript
   1. 一种面向对象的脚本编程语言
   2. 用于增加交互特性
   3. 基础结构
      1. 静态标准化
      2. 弱标准化
      3. 通过存储器自动调节设备
      4. 模型编程
      5. 将函数也当成是类（对象）的属性
   4. 结构
      1. 核心是ECMAScript
      2. BOM
      3. DOM
3. ECMAScript
   1. ES6相较于ES5新增内容：
      1. let 关键字。let关键字声明的变量仅在代码块内有效，且只能声明一次
      2. const相当于静态变量，且声明的同时必须赋值
      3. 多了symbol数据类型，类似唯一的const
      4. 解构赋值，用于数组或对象
      5. 新增map和set
      6. 箭头函数（类似java的labmda表达式），格式为 参数（参数如果有多个，要用括号）=>函数体
      7. ES6允许对象的属性直接写变量，这时候属性名是变量名，属性值是变量值。方法名也可以简写
      8. 从语法上说，Promise 是一个对象，从它可以获取异步操作的消息。十分复杂
      9. Class类。之前只有对象这个概念。能extends，能像java一样用构造函数实例化
      10. 遍历多了一个for…of
      11. 可变长度的参数
      12. 这个 … 操作符（也被叫做延展操作符 － spread operator）已经被 ES6 数组 支持。它允许传递数组或者类数组直接做为函数的参数而不用通过apply。
      13. 图形用户界面, 文本, 应用程序

          描述已自动生成
   2. ES7新增：
      1. Array.prototype.includes()用来判断一个数组是否包含指定的值，包含返回true，不包含返回false
      2. \*\* 与Math.pow(..)等效的计算结果。2\*\*10是2的10次方
4. AJAX与HTTP请求的同步和异步处理
   1. AJAX不用重新加载页面的情况下，部分地更新页面

文本

中度可信度描述已自动生成

* 1. [处理例子](https://www.huaweicloud.com/articles/a26205a80d3c4530f8d9954938ac2a00.html)
  2. XMLHttpRequest 是一个内建的浏览器对象，它允许使用 JavaScript 发送 HTTP 请求。[链接](https://zh.javascript.info/xmlhttprequest)

1. JQuery
   1. jQuery 是一个 JavaScript 函数库。
   2. jQuery 语法是通过选取 HTML 元素，并对选取的元素执行某些操作
   3. 菜鸟[链接](https://www.runoob.com/jquery/jquery-syntax.html)
   4. [实现AJAX](https://www.cnblogs.com/starof/p/6434791.html)
2. SuperAgent
   1. SuperAgent 是一个轻量的,渐进式的 ajax api
3. CGI
   1. 定义：由客户在服务端调用程序的技术
   2. 数据通过HTTP请求发送，由游览器生成
   3. 通常由程序的URL调用
   4. 每一个请求由单独的程序CGI进程加工
   5. Stdin以及stdout帮助与网络服务器交互
4. FastCGI
5. PHP
   1. 脚本语言
   2. 用;区分句，所以允许同行多句（不推荐，特别难看）
   3. $声明变量

实验要求：

1. 创建一个在规定范围内定义了坐标点的PHP-script。
2. 创建一个给PHP-script给HTML页面传输资料进行操作
3. 参数R由HTTP请求传入
4. Script应当完成资料的管理并以一个含有参数以及计算结果的表格返回给HTML页面
5. 上一个结果应当保存在请求中并且表示在表格中
6. 除此之外，回应应当包含当前时间以及工作时长
7. 对于文本以及图片元素的排列必须使用表格布局
8. 资料形式应当通过POST请求发送
9. 表格的处理形式应当单独放到一个文件中
10. 在使用CSS时应当表现出元素的选择器，子元素选择器，伪类选择器，标识符选择器以及类似于继承和级联的CSS特点
11. HTML页面应当具有“帽子”，包含了学生姓名，学号，вариант。在“帽子”的形成过程中必须用到清楚的serif，他的颜色以及尺寸当以级联样式表显示
12. 输入元素的填充必须以像素为单位指定。
13. 该页面必须包含一个 JavaScript 脚本，用于验证用户在表单字段中输入的值。任何无效的值（例如点坐标中的字母或负半径）都应该被屏蔽。

URI – 资源唯一标识，用于标志某一互联网资源的标识串 – 允许用户对网络的资源通过特殊协议进行交互。可以是URL或URN，或两者都是。

格式：[协议名]://[用户名]:[密码]@[服务器地址]:[服务器端口号]/[路径]?[查询字符串]#[片段ID]

UPL – 允许定义资源位置（资源在web上的唯一地址）

格式：protocol :// hostname[:port] / path/ [;parameters][?query]#fragment

URN – 包含了资源的唯一名称（与地址无关的标识方式）