1. Js
2. String
3. match() 找到一个或多个正则表达式的匹配
4. replace() 替换与正则表达式匹配的子串
5. Split() 把字符串分割为子字符串数组
6. Concat() 连接两个或多个字符串
7. 数组
8. concat() 合并数组
9. Join() 用数组元素组成字符串
10. Pop() 删除数组中最后一个元素
11. Push() 在数组末尾添加元素
12. Reverse() 顺序反转排序
13. Sort() 数组排序
14. Ts
15. Interface

interface IPerson {

firstName:string,

lastName:string,

sayHi: ()=>string

}

var customer:IPerson = {

firstName:"Tom",

lastName:"Hanks",

sayHi: ():string =>{return "Hi there"}

}

1. Class

class Car {

// 字段

engine:string;

// 构造函数

constructor(engine:string) {

this.engine = engine

}

// 方法

disp():void {

console.log("发动机为 : "+this.engine)

}

}

interface:接口只声明成员方法，不做实现。

class:类声明并实现方法。

1. Lua
2. Table



1. Jquery
2. Css()

$(selector).css(name) 返回第一个匹配元素的 CSS 属性值

$(selector).css(name,value)

$(selector).css(name,function(index,value)) 设置属性值

1. removeClass(“intro”) 移除“intro”的类
2. Html(“content”) 改变内容为“content”
3. Prop()

prop() 方法设置或返回被选元素的属性和值。

当该方法用于返回属性值时，则返回第一个匹配元素的值。

当该方法用于设置属性值时，则为匹配元素集合设置一个或多个属性/值对。

返回属性的值：

$(selector).prop(property)

设置属性和值：

$(selector).prop(property,value)

使用函数设置属性和值：

$(selector).prop(property,function(index,currentvalue))

设置多个属性和值：

$(selector).prop({property:value, property:value,...})

1. removeAttr() 从被选元素中移除属性。

Eg:$("p").removeAttr("id")

1. Siblings() 返回被选元素的所有同级元素。

Eg: $("li.start").siblings() 返回带有类名 "start" 的每个 <li> 元素的所有同级元素

1. Text() 设置被选中元素的内容
2. On() on() 方法在被选元素及子元素上添加一个或多个事件处理程序。

Eg: $("p").on("click",function(){

alert("段落被点击了。");

});

1. Append() 在被选元素的结尾（仍然在内部）插入指定内容.
2. Jquery 选择器
3. #id $("#lastname") id="lastname" 的元素
4. .class $(".intro") 所有 class="intro" 的元素
5. $("p") 所有<p>元素
6. Jquery ajax
7. $.ajax(url,[settings])

url:一个用来包含发送请求的URL字符串。

settings:AJAX 请求设置。所有选项都是可选的。

$.ajax({

//url

url: 'http://127.0.0.1:8000/jquery-server',

//参数

data: {a:100, b:200},

//请求类型

type: 'GET',

//响应体结果

dataType: 'json',

//成功的回调

success: function(data){

console.log(data);

},

//超时时间

timeout: 2000,

//失败的回调

error: function(){

console.log('出错啦!!');

},

//头信息

headers: {

c:300,

d:400

}

});

1. $.get(url,[data],[fn],[type])

url:待载入页面的URL地址

data:待发送 Key/value 参数。

callback:载入成功时回调函数。

type:返回内容格式，xml, html, script, json, text, \_default。

$.get("test.php", { name: "John", time: "2pm" },

function(data){

alert("Data Loaded: " + data); });

1. $.getJSON(url,[data],[callback])

url:发送请求地址。

data:待发送 Key/value 参数。

callback:载入成功时回调函数。

1. $.post(url,[data],[callback],[type])

url:发送请求地址。

data:待发送 Key/value 参数。

callback:发送成功时回调函数。

type:返回内容格式，xml, html, script, json, text, \_default。

get请求和post请求区别：https://www.cnblogs.com/jinaczg/p/9441709.html

1. Turbo.lua
2. requesthandler:入口点
3. RequestHandler:prepare()

如果您想在类初始化后执行某些操作，请重新定义此方法。

1. RequestHandler:get(url) HTTP GET 请求处理程序
2. RequestHandler:post(url) HTTP POST 请求处理程序
3. requesthandler:输入
4. RequestHandler:get\_argument(name, default, strip)

name ( String ) – 要获取的参数的名称。

default ( String ) – 如果未设置参数，则返回默认值。

strip ( Boolean ) – 从字符串的头部和尾部删除空格。

返回类型：string

1. RequestHandler:get\_arguments(name, strip)

返回具有相同名称的多个参数值。

返回类型：table

1. RequestHandler:get\_json(force)

以表格形式返回 JSON 请求数据

如果 force 设置为 true，则无论内容类型标头如何，都将解析所有请求。

返回类型：table/nil

1. requesthandler:输出
2. RequestHandler:write(chunk)

将给定的块写入输出缓冲区。如果给定的块是 Lua 表，它将自动字符串化为 JSON。

Chunk:string

1. RequestHandler:set\_status(code)

设置 HTTP 响应头的状态代码

Code:number

1. RequestHandler:get\_status()

获取 HTTP 响应头的当前状态代码

1. RequestHandler:redirect(url, permanent)

将客户端重定向到另一个 URL。

Url:string

Permanent:bool 将此标记为永久重定向或临时

1. error(turbo.web.HTTPError(400, "Could not find item in store"))

结果是状态码设置为 400 并且消息作为请求的正文发送

1. StaticFileHandler类

本地文件系统上文件的静态文件处理程序。在初始处理程序之前不会检查路径。如果文件是有效路径，则该文件将缓存在内存中。

local app = turbo.web.Application:new({

{"^/$", turbo.web.StaticFileHandler, "/var/www/index.html"}, //文件

{"^/(.\*)$", turbo.web.StaticFileHandler, "/var/www/"} //目录

})

1. application类

Application 类是组成 Web 应用程序的请求处理程序类的集合。

local app = turbo.web.Application:new({

{"/hello", HelloNameHandler}

})

Application:listen(port, address)

Port:启动http服务的端口

Address：string/number,将服务器绑定到的地址

IOLoop 是一个级别触发的 I/O 循环，额外支持超时和时间间隔回调

1. 文件函数
2. HTML中的& nbsp; & ensp; & emsp;等6种空格标记

<https://blog.csdn.net/weixin_30325971/article/details/95124748>

1. Typescript之class 中 constructor 方法 和 super 的作用

<https://www.10qianwan.com/articledetail/366249.html>

<https://www.jianshu.com/p/2f13ecb9a109>

1. Encode,decode

Base64.encode(data) 将字符串编码为Base64

Base64.decode(data) 将Base64解码为字符串

<https://segmentfault.com/a/1190000021624542>

1. Replace()方法

replace() 方法用于在字符串中用一些字符替换另一些字符，或替换一个与正则表达式匹配的子串。

1. hasOwnProperty

<https://www.cnblogs.com/yuqingfamily/p/5789568.html>

1. includes（“str”）

查找字符串中是否包含str

1. Util
2. Lua io.popen()

<https://www.jianshu.com/p/8ec58230633a>

1. Lua os.execute()

<https://www.jianshu.com/p/c3c70d046bfc>

<http://svn/websvn/wsvn/Main/tools/?#a2116a7c36c6988e4c7ec9da4e28a64c3>

1. Jquery.validate

<https://jqueryvalidation.org/documentation/>

<https://www.cnblogs.com/zhukaixin/p/10496474.html>

1. Jquery.validationEngine

<https://github.com/posabsolute/jQuery-Validation-Engine>

1. Jquery.ui

<https://jqueryui.com/>

<https://www.jqueryui.org.cn/tutorial/35.html>

1. jquery.blockUI.min.js

<http://bookshadow.com/weblog/2014/09/26/jquery-blockui-js-introduction/>

1. jquery.ui.touch-punch.min.js

<https://www.cnblogs.com/dreamzhiya/p/4933903.html>

1. <HTMLTableElement | null>document.getElementById(pgeObj.objName)

<https://developer.mozilla.org/en-US/docs/Web/API/HTMLTableElement/deleteRow>

16 . HTML DOM appendChild()

appendChild() 方法向节点添加最后一个子节点。

document.getElementById("myList").appendChild(newListItem);

17 . HTML DOM selected属性

selected 属性可设置或返回选项的 selected 属性的值。

该属性设置选项的当前状态，如果为 true，则该选项被选中。该属性的初始值来自 <option> 的 selected 属性。

<select>

<option id="apple">Apple</option>

<option id="orange">Orange</option>

<option id="pineapple" selected="selected">Pineapple</option>

<option id="banana">Banana</option>

</select>

1. checkGroupAccess()
2. Javascript fs模块

<http://javascript.ruanyifeng.com/nodejs/fs.html>

1. Jquery keyup()和keydown()事件方法

当按键按下或者松开时，触发事件回调函数。

$("input").keyup(function(){

$("input").css("background-color","#D6D6FF");

});

1. Jquery select()事件

当 textarea 或文本类型的 input 元素中的文本被选择时，会发生 select 事件。

select() 方法触发 select 事件，或规定当发生 select 事件时运行的函数。

$(selector).select() 对象被选中

$(selector).select(function) 对象被选中后执行回调函数

1. Javascript process.argv process.exit

<https://blog.csdn.net/zxj0904010228/article/details/83443585>

<http://nodejs.cn/api/process/process_argv.html>

<https://nodejs.org/docs/latest/api/process.html>

1. 定时器函数 setInterval 和setTimeout

<https://zhuanlan.zhihu.com/p/60731659>

<https://www.thinbug.com/q/49091543>

1. 浏览器解析执行过程

<https://www.cnblogs.com/dujuncheng/p/6850373.html>

1. Lua io.open,io.read

<https://www.jianshu.com/p/250386662d33>

<https://www.jianshu.com/p/8e90c1500eb9>

1. Lua string.sub()

<https://www.jianshu.com/p/8809b88ed087?nomobile=yes>

1. csrf攻击，cookie,session

<https://blog.csdn.net/besmarterbestronger/article/details/102544093>

1. Lua 模式匹配

<https://blog.csdn.net/yupu56/article/details/78115470>