



MONASH
University

程序代写代做 CS编程辅导

FIT1050 Web Fundamentals



Client-Side Scripting With JavaScript

WeChat: estutores

Assignment Project Exam Help

Email: tutores@163.com

QQ: 749389476

<https://tutores.com>

Week 9

Copyright Warning

Commonwealth of Australia Copyright Act 1968

程序代写代做 CS编程辅导



Warning

This material has been reproduced and communicated to you by or on behalf of Monash University in accordance with section 113P of the Copyright Act 1968 (the Act).

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

QQ: 749389476

<https://tutorcs.com>

Learning objectives

程序代写代做 CS编程辅导

What is JavaScript?

- The history and many names of JavaScript
- Understand what JavaScript is for



Language features

WeChat: cstutorcs

- Terminology and syntax rules
- Basic event handling model

Assignment Project Exam Help

Email: tutorcs@163.com

Using data and functions

QQ: 749389476

- Variables and data types
- Functions

<https://tutorcs.com>

程序代写代做 CS编程辅导



WeChat: cstutorcs

What is JavaScript?

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

JavaScript is not Java!

Developed in 1995 by Brendan Eich for Netscape, originally named "Mocha".

程序代写代做 CS编程辅导



- Eventually shipped as **LiveScript**
- Renamed to **JavaScript** to capitalise on Java's success.
- Microsoft develops a competing implementation named **JScript**

WeChat: cstutorcs

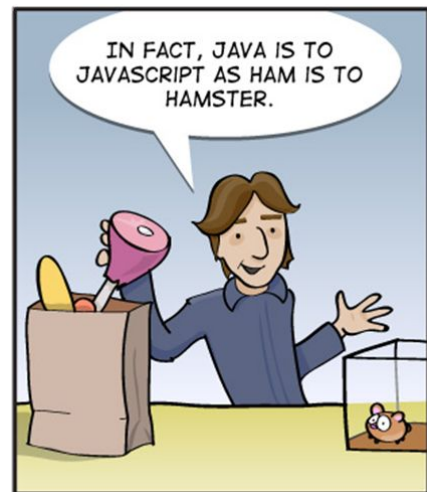
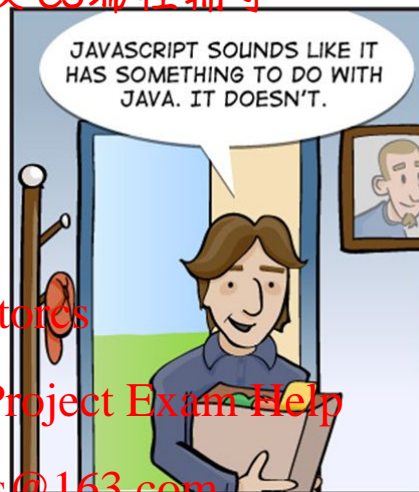
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Standardised by ECMA in 1996 using the name **ECMAScript**.



***The ECMAScript name sounds like a skin disease and won't ever catch on...
... Stop trying to make fetch happen.***

- Brendan Eich, 2018

Continued evolution of JavaScript

程序代写代做 CS编程辅导

The development of the ECMAScript standard continues:



- **ES1-3** Core language and standardisation improvements.
- **ES4** Abandoned due to development and compatibility concerns.
- **ES5** "ECMAScript 2009" - major revision includes new "strict mode".
- **ES6** "ECMAScript 2015" - major revision with wide browser support.
- **ECMAScript 2016-2022+** Ongoing yearly feature update versions.

WeChat: [cstutorcs](#)

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Unlike W3C standards, modern JavaScript features can break code in older browsers. Web developers are responsible for implementing compatibility.

JavaScript is...

JavaScript is designed as a single-threaded client-side scripting language

程序代写代做 CS编程辅导



- **Single-threaded**

- Commands are executed one at a time from a single call stack.

WeChat: cstutorcs

- **Client-side**

- Runs on the client - in the user's web browser.

Assignment Project Exam Help

- **Scripting language**

Email: tutorcs@163.com

- Programming languages are used to make standalone programs.
- Scripting languages run <https://tutorcs.com> to control existing programs.

QQ: 749389476

<https://tutorcs.com>

Well, actually... JavaScript can also run server-side using Node.js!

JavaScript as a client-side language

程序代写代做 CS编程辅导

JavaScript code is downloaded to the user's computer to be executed in the browser.

This has advantages compared to code on a remote server:

- Can process user interaction in real-time.
- Can execute code without using server resources.
- Can update the browser contents without waiting for a server.

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

Running client-side means JavaScript cannot perform some tasks:

QQ: 749389476

<https://tutorcs.com>

- User logins (database of users exists only on the server)
- Secure operations (code loaded into browser can be inspected by users)
- Store data on the server (e.g. posting to content online for others to access)

程序代写代做 CS编程辅导



Language Features and Syntax

WeChat: cstutors

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Syntax example

JavaScript uses a C-like syntax.

- Executable statements
- Statement blocks
- Variables for data storage
- Functions for code reuse
- Objects that have their own
 - Variables (properties)
 - Methods (functions)
- Conditional/iterative control flow

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

```
var i;  
var end = 10;
```

```
for ( i = 0; i < end; i++ ) {  
    console.log( i * 2 );  
}
```

If you have some programming experience, core concepts should be familiar.

Statements

程序代写代做 CS编程辅导

- Any single command instruction is a **statement**.
- Statements that can evaluate to produce a value are called **expressions**.



```
var a = 1;
```

// statements can be short and simple

WeChat: cstutorcs

```
if ( a == 1 ) {
```

// statement blocks have one or more commands

Email: tutorcsl@163.com

```
document.querySelector( 'body' ).style.color = '#ff0000';
```

```
document.querySelector( 'body' ).innerHTML = 'Hello';
```

QQ: 549389476

```
} else {
```

// the start of another statement block

https://tutorcs.com

```
document.querySelector( 'body' ).innerHTML = 'Goodbye';
```

```
}
```

Case sensitivity

All JavaScript keywords and identifiers are case-sensitive!

程序代写代做 CS编程辅导



```
console.log();
```

```
Console.log();
```

```
console.LOG();
```

// outputs a log message

// Error: Console is undefined

WeChat: cstutorcs

// Error: LOG is not a function

Assignment Project Exam Help

Email: tutorcs@163.com

```
var firstname = "Will";
```

// this is a variable

```
var firstName = "Will";
```

// this is a different variable

```
VAR firstname = "Will";
```

<https://tutorcs.com>

// Error: unexpected identifier

QQ: 749389476

Whitespace and semicolons

程序代写代做 CS编程辅导

Some guidelines for readable code:

- Finish statements with a semicolon (actually somewhat optional)
- Start complete statements on a new line (for readability)
- Indent code within block (for readability)

WeChat: cstutorcs

Assignment Project Exam Help

However, this code written as a single line is perfectly valid:

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

```
var a=1;if(a==1){document.querySelector('body').style.color
='#ff0000';document.querySelector('body').innerHTML='Hello'}
else{document.querySelector('body').innerHTML='Goodbye'}
```

Comments

Inline and block comments are both supported:

程序代写代做 CS编程辅导



```
// Single line comments the next newline
```

```
var a = 1; // comments can start after code
```

```
// var b = 2; this entire line of code is disabled
```

```
/* Block comments
```

```
can span across
```

```
multiple lines */
```

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Comments are useful for internal documentation and testing.

JavaScript's event loop

程序代写代做 CS编程辅导

The browser constantly listens for all event messages using an event loop.



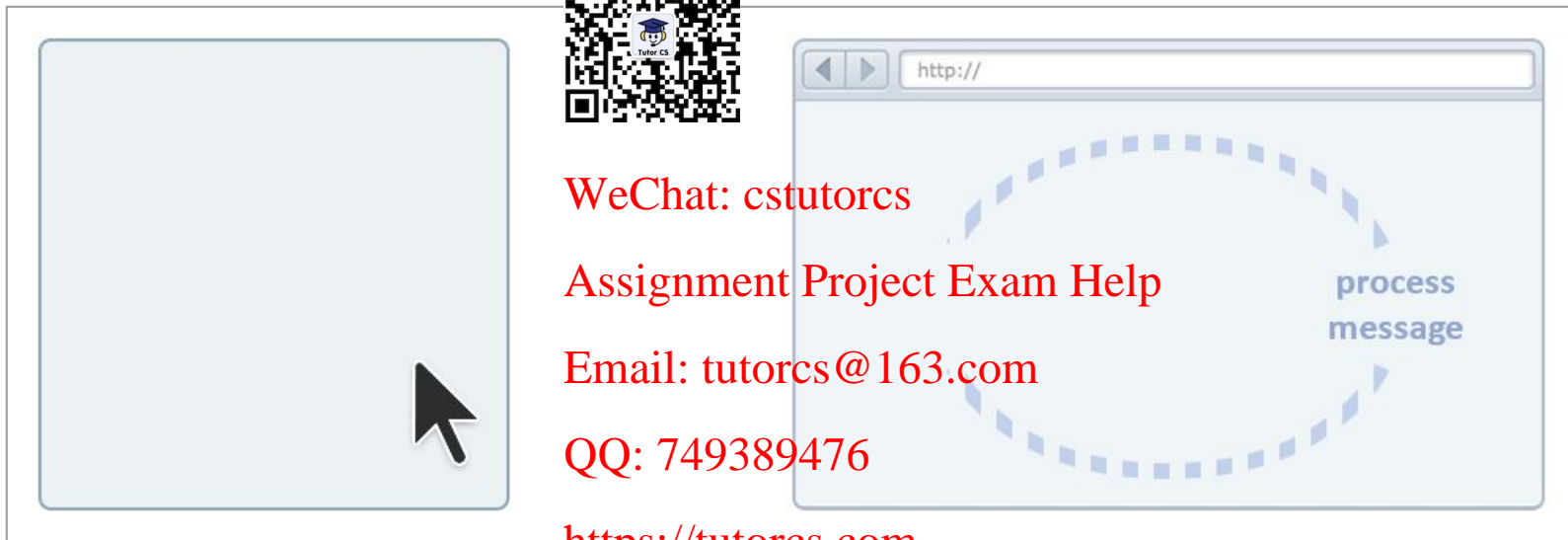
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutores.com>



We can attach listeners for specific events to HTML elements, allowing elements to react to messages by running code.

Event-based programming

Most JavaScript code is used to add interactions to a webpage.

程序代写代做 CS编程辅导



Typically:

1. Select an element **A** to interact with.
2. Select an element **B** to modify.
3. Wait for an interaction on element **A** and call a function when it happens.
4. The function modified element **B**.

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

```
var thingA = document.  
    querySelector("#A");
```

```
var thingB = document.  
    querySelector("#B");
```

```
thingA.addEventListener(  
    "click", updateB);
```

```
function updateB() {  
    thingB.style.color = 'red';  
}
```


程序代写代做 CS编程辅导



Working with Data And Variables

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Rules for naming variables

程序代写代做 CS编程辅导

Names must begin with either a:

- Letter (case-sensitive)
- _ (underscore) or \$



After the first letter, names can contain:

- Letters (case-sensitive)
- Numbers
- Underscores and dollar signs

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Identifiers cannot be named using reserved keywords.

Data can be stored in variables

程序代写代做 CS编程辅导

You should declare each variable the first time it is used.



```
var a;
```

```
var b = 1;
```

```
a = 2;
```

```
console.log( a, b );
```

```
var c, d, e;
```

```
var f = 2, g = 3, h = 4;
```

a is declared, but undefined

// b is declared, and defined as 1

WeChat: cstutorcs

// a is now defined as 2

Assignment Project Exam Help

// Output: 2, 1

Email: tutorcs@163.com

QQ: 749389476

// c, d, e declared, but undefined

<https://tutorcs.com>

// f, g, h declared and defined

ES6 block-scoped variables

程序代写代做 CS编程辅导

In modern browsers, the **let** keyword declarations variable that can be scoped to **any** statement block - **not just function**



```
for ( var i = 0; i < 10; i++ ) {  
}
```

WeChat: cstutorcs

```
console.log( i );
```

Assignment Project Exam Help

// Output: 10

Email: tutorcs@163.com

```
for ( let i = 0; i < 10; i++ ) {  
}
```

QQ: 749389476

<https://tutorcs.com>

```
console.log( i );
```

// i is not defined (no longer exists)

ES6 block-scoped constants

程序代写代做 CS编程辅导

In modern browsers, variables declared with the **const** keyword cannot be updated after the initial declaration:



```
let limit = 10;
for ( let i = 0; i < limit; limit++ ) { // infinite loop!
}
```

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

```
const limit = 10;
for ( let i = 0; i < limit; limit++ ) { // exit with error
}
```

Primitive data types

JavaScript provides just a few primitive data types, which are dynamically handled.

Any non-constant variable can be assigned with a value or any type.

程序代写代做 CS编程辅导



- **Strings**

`""`

`'a'`

`"1"`

`'abc'`

`"abc def"`

WeChat: cstutorcs

- **Numbers**

`0`

`1`

`-10`

`12345`

`1.2345677`

Assignment Project Exam Help

Email: tutorcs@163.com

- **Booleans**

`false`

`true`

QQ: 749389476

<https://tutorcs.com>

- **Null (no value)**

`null`

Automatic type coercion

程序代写代做 CS编程辅导

Operations with different data types, result in automatic type coercion.



```
var a = 1 + 2;
```

// no coercion, a = 3

```
var b = 1 + '2';
```

// implicit coercion, b = "12"

```
var c = '1' + 2;
```

// implicit coercion, c = "12"

```
var d = Number( '1' ) + 2;
```

// explicit coercion, d = 3

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

Automatic coercions can be dangerous, but also convenient:

<https://tutorcs.com>

```
console.log( 'The value of variable a is: ' + a );
```

Truthy and falsy

程序代写代做 CS编程辅导

The following values are treated as falsy (can coerce to Boolean **false**):

- Empty strings
- Number zero
- Boolean false

""

0

false



0.0

-0.0

WeChat: cstutorcs

Assignment Project Exam Help

By definition **null**, **undefined** and **NaN** are **falsy**, but will require special handling.

Email: tutorcs@163.com

QQ: 749389476

All other values are interpreted as **truthy** (coerce to Boolean as **true**).

<https://tutorcs.com>

```
if ( 'hello' && 5 && true ) { console.log( 'truthy' ) }
```


程序代写代做 CS编程辅导



Creating Functions to Reuse Code

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Declaring and calling functions

程序代写代做 CS编程辅导

Functions are often declared with a identifier name.



```
function sayHello() {  
    return 'hello';  
}
```

// sayHello declared

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

// executed by calling

```
sayHello();
```

QQ: 749389476

<https://tutorcs.com>

This allows the code within a function to be re-used easily - especially important for complex functions containing logic

Camel-case identifier names are preferred

程序代写代做 CS编程辅导

You can use a mix of uppercase and lowercase letters, but conventions exist for general naming style to promote consistency.



```
function addNumbers() { ... } // camel-case is preferred
```

WeChat: cstutorcs

```
function AddNumbers() { ... } // legal, looks like a class
```

Assignment Project Exam Help

Email: tutorcs@163.com

```
function aDdNuMbErS() { ... } // legal, but please don't!
```

QQ: 749389476

<https://tutorcs.com>

Remember that JS is case-sensitive - **apply case consistently!**

Variables can be scoped to a function

程序代写代做 CS编程辅导

Variables declared using **var** outside of functions are globally accessible. When the declaration is within a function, the variable is accessible within the function only.



```
function sayHello() {  
    var a = 'hello';  
    return a;  
}
```

```
sayHello();  
console.log( a );
```

WeChat: cstutorcs

// Can also use **let** for block-scope

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

// Output: "hello"
<https://tutorcs.com>

// ReferenceError: a is not defined

Functions can import data

程序代写代做 CS编程辅导

Functions can import external data into parameters. Parameters are scoped variables that are defined as part of the function.



```
function addNumbers( a, b ) {           // declare some parameters
    return a + b;                       // use them like variables
}
```

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

```
addNumbers( 1, 2 );
```

QQ: 749389476

// pass arguments into parameters

<https://tutorcs.com>

This allows a function to be reused and execute using different data on each call.

Defining function as expressions

程序代写代做 CS编程辅导

This is practically identical in most situations:



```
const addNumbers = function(a, b) { // declare the function
    return a + b;
}
```

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

```
addNumbers( 1, 2 ); // call the function
```

This example uses **const** to ensure that the function definition can never be accidentally overwritten. However, **var** and **let** can also be used.

ES6 arrow function expressions

程序代写代做 CS编程辅导

A newer compact function expression syntax can be used in modern web browsers:



```
function addNumbers( a, b ) {  
    return a + b;  
}
```

// traditional declaration

WeChat: cstutorcs

Assignment Project Exam Help

```
let addNumbers = ( a, b ) => {  
    return a + b;  
}
```

// arrow function expression

QQ: 749389476

<https://tutorcs.com>

```
let addNumbers = ( a, b ) => a + b;  
  
// implicit return
```

Next week

程序代写代做 CS编程辅导

- Mobile web design and development.
- Implementing basic mobile responsive layout using CSS.



Self-study this week

WeChat: cstutorcs

- Choose your topic and template for Assignment 3.
- Work on research and planning for your website construction.

Assignment Project Exam Help

Email: tutorcs@163.com

Reminders

QQ: 749389476

- Participation Milestone 4 takes place during next week's lab class.

<https://tutorcs.com>