

程序代写代做 CS编程辅导

COMP6080



Front-End Programming

Week 10 — The End

WeChat: cstutorcs

Sidney Pham
Assignment Project Exam Help
sidney.pham@unsw.edu.au

Email: tutorcs@163.com

UNSW

QQ: 749389476

Week 10, Term 3 2020

<https://tutorcs.com>

The End

程序代写代做 CS编程辅导



We've made it to the end of content! Personally, it's the last week of tutoring for me, which feels like it deserves reflection.

Thanks for collectively being a really cool class!

Speaking of reflection, my experience is up — help make this course better next time!

Oh, also, I have no idea what happens to my slides on Gitlab once I graduate, maybe download them if you want to keep them.

<https://tutorcs.com>

The Tutorial

程序代写代做 CS编程辅导

Tutorial 10 is an opportunity for the tutor and their students to highlight particular topics that students feel might need a bit more exploration, particularly in the exam that is coming up. A number of topics have been brought up that you may want your tutor to go through:



1. HTML

WeChat: cstutorcs

2. CSS — Selectors, Formatting, Layouts, Dev Tools

Assignment Project Exam Help

3. NodeJS — Basics

Email: tutorcs@163.com

4. Javascript in browser — Importing, DOM manipulation, forms, events, local storage

QQ: 749389476

5. ReactJS — Components, hooks, routing

https://tutorcs.com

6. UI/UX principles

7. Accessibility principles

8. Testing principles

Giving You Some Thinking Time...

程序代写代做 CS编程辅导



WeChat: cstutorcs

Please think of things you're a little confused about in this course!
But in the meantime, I just wanted to cover one fun thing today
— closures!

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

TL·DR

程序代写代做 CS编程辅导



You can think of a closure as a tuple: (function, environment).

WeChat: cstutorcs

An **environment** is a mapping from variables to values. That is, it's basically a function that gives you values for variables.

Assignment Project Exam Help

That's all the remembering you need to do, but it probably won't make much sense. Let's try and explain...

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

A Binary Adder

程序代写代做 CS编程辅导



Let's create a function that we can use like
`binaryAdd(2)(3) === 5`. An important thing to note is that we
have **first-class functions**! (`binaryAdd` is a function that returns a
function.)

Assignment Project Exam Help

```
const binaryAdd = x => (y => x + y)
```

(The `=>` operator is right-associative, so you don't actually need
those parentheses; it's there to emphasise that we're returning a
function.)

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Using It

程序代写代做 CS编程辅导

We can use it like:

```
const binaryAdd = (x) => y => x + y
const add2 = binaryAdd(2);
console.log(add2(5)) // 7
```

WeChat: cstutorcs

Calling `binaryAdd(2)` returns a function that takes a number `y` and returns `x + y`. But notice that `x` is only available within the scope of `binaryAdd`.

Assignment Project Exam Help

Email: tutorcs@163.com

Calling `add2(5)` will bind `y` to 5 and then try to compute `x + 5`. This will try and find the value for `x` by performing **scope resolution** (checking the current environment, then all the outer environments).

QQ: 749389476

<https://tutorcs.com>

Closures

程序代写代做 CS编程辅导



What happens in languages with closures (like JavaScript) is that `binaryAdd(2)` returns a closure (i.e. (function, environment)), not just a function. So, the lifetime of `x` is extended beyond the lifetime of `binaryAdd`.

Assignment Project Exam Help

For example, we might do something like

```
window.onClick = () => alert(name);
```

 Whenever

`window.onClick` gets called, the value for `name` will be found in the closure!

QQ: 749389476

<https://tutorcs.com>

That's Pretty Much It

程序代写代做 CS编程辅导



It's honestly not the most important thing to know; it's hopefully already intuitively obvious. However, I think understanding *why* things work lets you talk about your code better (a very important thing).

WeChat: [tutorcs](#)

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

There is More..

程序代写代做 CS编程辅导



One thing to notice in closures kind of let us do **encapsulation**. Before, the `x` variable was available in the global scope, but we could run `add2(5)` successfully! We've hidden some state in our closure.

WeChat: [cstutorcs](#)

See the **Emulating private methods with closures** section of the MDN page for an example of how you can use functions as a simplified class!

Assignment Project Exam Help

Email: tutorcs@163.com

Essentially, our closure is *one* function with as much encapsulated state (the variables we *close over*) as we like.

QQ: 749389476

<https://tutorcs.com>

Bonus: Currying

程序代写代做 CS编程辅导



A common technique in functional programming (which you can totally do in JavaScript) is **currying**.

Notice how we didn't do

`const uncurriedAdd = (x, y) => x + y`, which would be called like `uncurriedAdd(2, 5)` instead of `binaryAdd(2)(5)`?

We've written a **curried** version, and now we can pass around **partially applied** functions. For example,

```
[1, 5, 2, 3].map(x => uncurriedAdd(2, x))
```

vs

```
[1, 5, 2, 3].map(binaryAdd(2))
```

Things to Cover

程序代写代做 CS编程辅导



In the likely event that out of things to discuss, here are some ideas:

- Semantic HTML elements;
- Doing a random assignment, 2 code review;
- Using a CSS framework like Bulma in React (rbx);
- Going through a lab question together;
- Trying to list UI/UX considerations on the fly;

<https://tutorcs.com>