

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

JavaScript, Web Frameworks and Visuals

BDAT 1004

Week 8

Announcements

- 50% of the class is completed!
- We have covered most of the “tough” material
- Problem Set 3 will be up next week and do last day of class
- Don't forget this weeks quiz!
- This lecture has a tutorial but no exercises
- Changes to the Syllabus

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



JS Browser Console

- All txt files on blackboard change to HTML
- I recommend using Chrome or FireFox
- However this can be done using any web browser

Assignment Project Exam Help

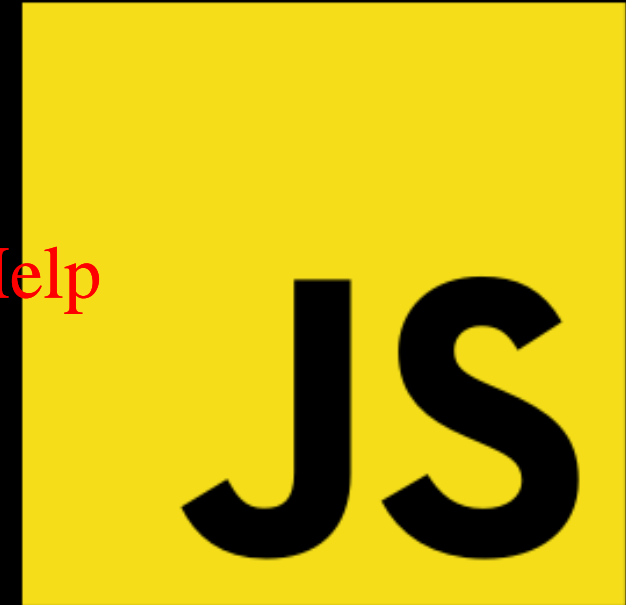
<https://powcoder.com>

Add WeChat powcoder



JavaScript

- A high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm
- Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications
- The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it



JavaScript Coding Conventions

Coding conventions are **style guidelines for programming**. They

Naming and declaration
rules for variables and
functions.

Rules for the use of
white space, indentation,
and comments.

Programming practices
and principles



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Coding conventions **secure quality**:

Improves code readability

Make code maintenance easier



Coding conventions can be documented rules for teams to follow,
or just be your individual coding practice.

JavaScript Coding Conventions

Variable Names

- **camelCase** for identifier names (variables and functions).
- All names start with a **letter**.

Spaces Around Operators

- Always put spaces around operators (= + - * /), and after commas:

Code Indentation

- Always use 4 spaces for indentation of code blocks:
 - ```
function toCelsius(fahrenheit) {
 return (5 / 9) * (fahrenheit - 32);
}
```

# JavaScript Coding Conventions

## Assignment Project Exam Help

### Statement Rules

- General rules for simple statements:
- Always end a simple statement with a semicolon.

<https://powcoder.com>

Add WeChat powcoder

Examples:

```
var values = ["John", "Doe", "50", "blue"];

var person = {
 firstName: "John",
 lastName: "Doe",
 age: 50,
 eyeColor: "blue"
};
```

# JavaScript Coding Conventions

## Assignment Project Exam Help

### Object Rules

- General rules for object definitions:
- Place the opening bracket on the same line as the object name.
- Use colon plus one space between each property and its value.
- Use quotes around string values, not around numeric values.
- Do not add a comma after the last property-value pair.
- Place the closing bracket on a new line, without leading spaces.
- Always end an object definition with a semicolon.

<https://powcoder.com>  
Example

Add WeChat powcoder

```
var person = {
 firstName: "John",
 lastName: "Doe",
 age: 50,
 eyeColor: "blue"
};
```



# JavaScript Coding Conventions

## Assignment Project Exam Help

Line Length < 80

- For readability, avoid lines longer than 80 characters.
- If a JavaScript statement does not fit on one line, the best place to break it, is after an operator or a comma.

<https://powcoder.com>

Example

```
document.getElementById("demo").innerHTML =
 "Hello Dolly.";
```

Add WeChat powcoder

# JavaScript Where To

## Assignment Project Exam Help

### The <script> Tag

- In HTML, JavaScript code must be inserted between <script> and </script> tags.

### External JavaScript

- Scripts can also be placed in external files

#### Example

<https://powcoder.com>

Add WeChat powcoder

```
<script>
document.getElementById("demo").innerHTML = "My First JavaScript";
</script>
```

#### Example

```
<script src="myScript.js"></script>
```

# JavaScript Variables

- All JavaScript **variables** must be **identified** with **unique names**.
  - These unique names are called **identifiers**.
  - Identifiers can be short names (like x and y) or more descriptive names (age, sum, totalVolume).
  - The general rules for constructing names for variables (unique identifiers) are:
    - Names can contain letters, digits, underscores, and dollar signs.
    - Names must begin with a letter
    - Names can also begin with \$ and \_ (but we will not use it in this tutorial)
  - Names are case sensitive (y and Y are different variables)
  - Reserved words (like JavaScript keywords) cannot be used as names
  - JavaScript Identifiers are case-sensitive

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# JavaScript Variables

- Declaring (Creating) JavaScript Variables
  - Creating a variable in JavaScript is called "declaring" a variable.
  - You declare a JavaScript variable with the **var** keyword:
- After the declaration, the variable has no value. (Technically it has the value of **undefined**)
  - To **assign** a value to the variable, use the equal sign:
- You can also assign a value to the variable when you declare it:

```
var carName;
```

```
carName = "Volvo";
```

```
var carName = "Volvo";
```

# JavaScript Arithmetic Operators

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

| Operator | Description                  |
|----------|------------------------------|
| +        | Addition                     |
| -        | Subtraction                  |
| *        | Multiplication               |
| /        | Division                     |
| %        | Modulus (Division Remainder) |
| ++       | Increment                    |
| --       | Decrement                    |

# JavaScript Assignment Operators

Assignment operators assign values to JavaScript variables.

| Operator        | Example             | Same As                |
|-----------------|---------------------|------------------------|
| =               | <code>x = y</code>  | <code>x = y</code>     |
| <code>+=</code> | <code>x += y</code> | <code>x = x + y</code> |
| <code>-=</code> | <code>x -= y</code> | <code>x = x - y</code> |
| <code>*=</code> | <code>x *= y</code> | <code>x = x * y</code> |
| <code>/=</code> | <code>x /= y</code> | <code>x = x / y</code> |
| <code>%=</code> | <code>x %= y</code> | <code>x = x % y</code> |

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# JavaScript String Operators

## Assignment Project Exam Help

- The + operator can also be used to add (concatenate) strings.

Example: <https://powcoder.com>

Add WeChat powcoder

```
var txt1 = "John";
var txt2 = "Doe";
var txt3 = txt1 + " " + txt2;
```

The result of txt3 will be:

John Doe

## Adding Strings and Numbers

### Assignment Project Exam Help

- Adding two numbers, will return the sum, but adding a number and a string will return a string:

<https://powcoder.com>

Add WeChat powcoder

Example

```
var x = 5 + 5;
var y = "5" + 5;
var z = "Hello" + 5;
```

The result of x, y, and z will be:

```
10
55
Hello5
```



## JavaScript Comparison Operators

| Operator | Description                       |
|----------|-----------------------------------|
| ==       | equal to                          |
| ===      | equal value and equal type        |
| !=       | not equal                         |
| !==      | not equal value or not equal type |
| >        | greater than                      |
| <        | less than                         |
| >=       | greater than or equal to          |
| <=       | less than or equal to             |
| ?        | ternary operator                  |

## JavaScript Logical Operators

| Operator | Description |
|----------|-------------|
| &&       | logical and |
|          | logical or  |
| !        | logical not |

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

## JavaScript Bitwise Operators

| OPERATOR | DESCRIPTION           | EXAMPLE | SAME AS     | RESULT | DECIMAL |
|----------|-----------------------|---------|-------------|--------|---------|
| &        | AND                   | 5 & 1   | 0101 & 0001 | 0001   | 1       |
|          | OR                    | 5   1   | 0101   0001 | 0101   | 5       |
| ~        | NOT                   | ~5      | ~0101       | 1010   | 10      |
| ^        | XOR                   | 5 ^ 1   | 0101 ^ 0001 | 0100   | 4       |
| <<       | Zero fill left shift  | 5 << 1  | 0101 << 1   | 1010   | 10      |
| >>       | Signed right shift    | 5 >> 1  | 0101 >> 1   | 0010   | 2       |
| >>>      | Zero fill right shift | 5 >>> 1 | 0101 >>> 1  | 0010   | 2       |

## Further Study

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

- [JavaScript Tutorial - http://www.w3schools.com/js/default.asp](http://www.w3schools.com/js/default.asp)
- [JavaScript Introduction - https://www.w3schools.com/js/js\\_intro.asp](https://www.w3schools.com/js/js_intro.asp)
- [JavaScript Best Practices - https://www.w3schools.com/js/js\\_best\\_practices.asp](https://www.w3schools.com/js/js_best_practices.asp)
- [JavaScript Mistakes - https://www.w3schools.com/js/js\\_mistakes.asp](https://www.w3schools.com/js/js_mistakes.asp)
- [JavaScript Versions - https://www.w3schools.com/js/js\\_versions.asp](https://www.w3schools.com/js/js_versions.asp)
- [JavaScript Performance - https://www.w3schools.com/js/js\\_performance.asp](https://www.w3schools.com/js/js_performance.asp)
- [JavaScript Hoisting - https://www.w3schools.com/js/js\\_hoisting.asp](https://www.w3schools.com/js/js_hoisting.asp)

# Web Application Frameworks

A web application framework (WAF) is a software framework that is designed to support the development of web applications including web services, web resources, and web APIs.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Web frameworks provide a standard way to build and deploy web applications on the World Wide Web.

Web frameworks aim to automate the overhead associated with common activities performed in web development.

For example, many web frameworks provide libraries for database access, templating frameworks, and session management, and they often promote code reuse

# Web Application Frameworks

There are hundreds of different frameworks, most languages have multiple frameworks to work with. Some frameworks you may be familiar with are React JS (used by Facebook) Angular JS (Netflix) Django (Instagram) and Ruby on Rails (Twitter)

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Since we mainly work with Python we will be covering Flask (a light weight version Django).

However you are more than welcome to choose any framework you wish for your final project

# Flask



Microframework for Python



Assignment Project Exam Help Designed for expansion

<https://powcoder.com>

Add WeChat powcoder



Includes only web server and templating engine



Easy to add new functionality via extensions

# Virtual Environment

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

A virtual environment is a tool that helps to keep dependencies required by different projects separate by creating isolated python virtual environments for them.

Imagine a scenario where you are working on two web based python projects and one of them uses a Django 1.9 and the other uses Django 1.10 and so on. Virtual environment can be really useful to maintain dependencies of both the projects.



# Virtual Environment

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

- By default, every project on your system will use these same directories to store and retrieve site packages (third party libraries).
- How does this matter? Now, in the above example of two projects, you have two versions of Django. This is a real problem for Python since it can't differentiate between versions in the "site-packages" directory. So both v1.9 and v1.10 would reside in the same directory with the same name.
- This is where virtual environments come into play. To solve this problem, we just need to create two separate virtual environments for both the projects.
- The great thing about this is that there are no limits to the number of environments you can have since they're just directories containing a few scripts.

# Getting Started

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

- Anaconda already comes with Flask but we should set up a virtual environment
- What virtualenv does is create a sandbox environment for you so you can install whatever python libraries you want without affecting anything else on your computer.
- When you delete that virtual environment, all those libraries go away like it never happened.
- This will make it much easier when you migrate your app to the cloud
- Go to the anaconda navigator RUN AS ADMIN
- Conda install virtualenv

File Edit Selection View Go Debug Tasks Help

flask-basic.py x

```
1 # Basic flask application
2
3 from flask import Flask
4 app=Flask(__name__)
5
6
7
8
9 @app.route('/')
10 def index():
11 return '<h1>Learning Flask</h1>'
12
13 @app.route('/user/<name>')
14 def user(name):
15 return 'Hello, %s' % format(name)
16
17
18
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Building a basic web app

# Building a basic web app

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

- The first section creates the Flask application. Flask will run on port 5,000 by default
- but it still needs to know what to do when a request comes in, so we can jump right into defining routes.
- In the first route, you can see the basic structure of the route definition. The definition uses a decorator with the @ sign to specify the route being defined. The slash means that Flask will serve this page from the root URL of the server. It will be served at local host colon 5,000 slash.
- Next, a function is defined called index which returns an HTML h1 element of learning Flask.
- The second route is more interesting. We're defining a variable based on the second part of the URL defined in using angle brackets. So when this URL is visited, the response will have that part of the path included as a variable for our function to work with.
- The function here is named user, and the name is passed along from that path variable. Note that we're using Python formatting to create this string along with the format function from the templating engine.

# Running the application

- Head back to the anaconda prompt
- Change the directory to where your application and requirements are
- Set up your virtual environment here by typing:
  - `virtualenv venv`
- Then we need to activate the environment:
  - `venv\Scripts\activate`
- Then we install the requirements
  - `pip install -r requirements.txt`
- Set `FLASK_APP=flask-basic.py`

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Bootstrap

- Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development.
- It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation and other interface components
- Bootstrap is a component of the Flask library and can be integrated into your app for a better appearance

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



# Charts

---

Charts are an excellent way to visualize data on your web application.

---

There are many add ons that can be integrated into flask to visualize your analysis

**Assignment Project Exam Help**

Such as Google Charts, Chart.js and Chartis.js

**<https://powcoder.com>**

**Add WeChat powder**

Most chart APIs require some cut and paste javascript and due to this there needs to be 2 parts to your python application, the main app and the HTML templates

---

I will show you a video tutorial on Google Charts

# Google Charts

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

The Google Chart API is an interactive Web service (now deprecated) that creates graphical charts from **user-supplied data**.

Google servers create a PNG image of a chart from data and formatting parameters specified by a user's HTTP request.

The service supports a wide variety of chart information and formatting. Users may conveniently embed these charts in a Web page by using a simple image tag.



# Using Charts

Set Up

First let's set up a file directory

Create

Create a project folder with a project name, I named mine flaskCharts

<https://powcoder.com>

Create

Inside this project folder create 2 more folders one called "static" and one called "templates"

Open

Now open Visual Studio Code (or whatever text editor you prefer)

# Configure Flask

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat [powcoder](https://powcoder.com) Flask

```
app = Flask(__name__)
```

Create the below **app.py** script(py is the extension to indicate Python script) where we import the flask module. This file should be created under the project root directory. Notice how we create flask instance.

## Configuring URLs

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

- Next, we will create main.py script that will define the URI and render a template file for displaying data on Google pie chart.
- Ideally the data should come from database or any other source and a chart should be generated based on real data. But for example purpose, here we are hard-coding the data to be displayed on pie chart.
- We use `http method GET` for displaying view. By default http method is `GET` if you do not specify http method.
- We use `render_template` function from flask to show the view. We pass the data as an argument to the `render_template` function.

## View – Template File

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

- In templates directory under the project's root directory, create a template file pie-chart.html with the content html file on Black Board.
- Now we will add Google chart API to display our data on chart. We will iterate dictionary items to build the pie chart.
- We are also checking whether data is string or not. If the data type is string then we quote around it otherwise we simply put the integer value.
- We configure chart options using options array with title and whether the pie chart should be rendered as 3D (`is3D = true`) or not (`is3D = false`). By default pie chart is not rendered as 3D (`is3D = false`).
- <https://developers.google.com/chart/interactive/docs/gallery/piechart>

# Running and Testing the Application

- Now navigate to the project's root directory and execute the command `python main.py` as shown in the below image, your server will start on default port 5000.

Assignment Project Exam Help

- If you want to change the port then you can change the line `app.run()` to `https://powcoder.com=50001`, where 5001 is the new port.

Add WeChat powcoder

- Hit the URL `http://localhost:5000/google-charts/pie-chart` to open the home page