Assignments

PE01 & PE02

```
<!DOCTYPE html>
<html lang="en" xml:lang="">

    <head>
        <title>My Starter Web page</title>
        <meta name="author" content="Thaddeus Thomas" />
        <img src="./IMG/background.jpg" content="background" />
        <!-- Local style sheet relative to workspace folder -->
        <link rel="stylesheet" href="./css/style.css" />
        <style>
            img {
                width: 150%;
            }
        </style>
        <!-- Local style sheet relative to this file -->
        <link rel="stylesheet" href="./css/style.css" />
    </head>

    <body>
        <div class="container"
                style="padding-top: 8px;
                       text-align: left;
                       font-size: 15px;
                       font-family: Source Sans Pro, Arial, sans-serif;">
            <div>
                <h2>Light Switch</h2>
                <img id="light" src="./IMG/PE02/lightOff.png" style="width: 100px"
/>
```

```
                    <script>
                        function toggleLightSwitch()
                        var imgLight = document.getElementById("light")
                        if (lightOff.match("OFF"))
                            imgLight.src() = "../code/thad/PE02/img/lightOff.png";
                        else
                            imgLight.src() = "../code/thad/PE02/img/PE02/lightOn.png";
                    </script>
                    <button type='button' id="OFF"
onclick="toggleLightSwitch();">ON/OFF</button>
                </div>
        </body>

</html>
```

PE03

```
Course: IS 312 – Web Design and Programming
Quarter: Fall - 2022
Your Name: Thaddeus Thomas
Assignment: PE03
```

Task # 1

```
// TypeScript
let firstName: string;
let lastName: string;
let fullName: string;
let age: number;
let ukCitizen: boolean;

firstName = "Rebecca";
lastName = "Smith";
age = 42;
ukCitizen = false;
fullName = `${firstName} ${lastName}`;

if (ukCitizen) {
  console.log(
    "My name is " +
      fullName +
      ", I'm " +
      age +
      ", and I'm a citizen of the United Kingdom."
  );
} else {
  console.log(
    "My name is " +
      fullName +
      ", I'm " +
      age +
      ", and I'm not a citizen of the United Kingdom."
  );
}
```

Task # 2

```
let x: number;
let y: number;
let a: number;

x = 5;
y = 7;
a = x + y;

console.log(a);
```

Task # 3

```
let randomNumbers: number;
let nextNumber: number;

for (let i = 0; i < 10; i++) {
  nextNumber = Math.floor(Math.random() * (100 - 1)) + 1;
  randomNumbers.push(nextNumber);
}

console.log(randomNumbers);
```

PE04

```
// Task 1-1
var slices: string[];
var pizzaTypes: string[];
var price: number[];
// Task 1-2
slices = new Array(
  "Cheese",
  "Meatlovers",
  "Supreme",
  "BBQ",
  "Pepperoni",
  "Pineapple & Ham"
);
pizzaTypes = ["omni", "meat", "veggie"];
price = [5, 10, 20];
// This is supposed to be the type or something IDK what the instructions want or
```

```
how to interpret them.
type pizza = {
  Name: string;
  Type: string;
  Price: number;
};
function getPizza(array) {
  for (var i = 0; i < array.length; i++) {
    console.log(array[i]);
  }
}
// Task 2-1
let transactions = [15, 20, 30, 10, 5];
var total = getPizza(transactions);
// Function that isnt set up right for logging prices
function writePrice(slices: string, price: number): void {
  console.log(`Price for ${slices}: $${price.toFixed(2)}`);
}
// Task 3-1
var total2 = transactions.reduce(function (a, b) {
  return b;
});
console.log(`Total is: $${total2}`);
```

PizzaShop

```

class Slice {
  name: string[] = ["Cheese", "Pepperoni", "MeatLovers", "Veggie", "BBQ"];
  type: string[] = ["Omni", "Meat", "Meat", "Veggie", "Meat"];
  price: number[] = [2, 3, 4, 1, 5];
}
console.log(Slice);

let names: string[] = [];
let transaction: number[] = [];

function writePrice(Slice: string, price: number): void {
  console.log(`Price for ${Slice}: $${price.toFixed(2)}`);
}
```

PE05

```

/*  Course:     IS 312 - Web Design and Programming
 *   Quarter:    Fall 2022-23
 *   Your Name:  Thaddeus Thomas
 *   Task #1:    Input Validation
 *   For this task you will need to create two functions:
 *   1. A function that creates a input prompt that requests the user to input
 *        a number between 1 and 20.
 *   2. A function to validate the input the user gave,
 *        to verify that the user’s input is valid and a number.
 */

// Task #1
function prompt(
  message?: string | number,
  _default?: string | number
): string | number;
const numRet = (x: (usrInput: string) => any): number => {
  return Number(x);
};

function inputNumber() {
  let usrInput = prompt("Please input a number between 1 & 20. ");
  const num1 = numRet;
  validInput(num1);
}

function validInput(input) {
  if (typeof input === "number" && input >= 1 && input <= 20) {
    console.log(`"${input} is a number between 1 & 20!"`);
    return true;
  } else {
    console.log(`"${input} that is not a valid answer"`);
    return false;
  }
}

inputNumber();
const num1 = numRet;

// Task #2
function taskTwo(p1: number = num1, p2: number = num1) {
  if (p1 <= 0) {
    return p2;
  } else {
    while (p1 > 0) {
      p2 = p1 * p2;
      taskTwo(p1, p2);
      p1 -= 1;
      return p1;
    }
  }
}

// Task #3 - Fibonacci
function fibonacci(i = inputNumber()) {
  var x: number = 0;
  var y: number = 1;
  var j = 0;
  while (i > j) {
    var alpha: number = x + y;
    x = y;
    y = alpha;
    j += 1;
    return i;
  }
}

fibonacci();
```

PE06

```
/*  Course:     IS 312 - Web Design and Programming
 *  Quarter:    Fall 2022-23
 *  Your Name:  Thaddeus Thomas
 *  Task #1:    Declaring an Interface
 *  Task #2     Instancing an Interface
 *  Task #3     Extending Interfaces
 *  Task #4     Learning And Challenges
 */

// Task #1
interface Pizza {
    type: string,
    slices: number,
    crust?: string;
};

let myPizza: Pizza = {
    type: 'cheese',
    slices: 8,
    crust: 'stuffed'
};

console.log(myPizza)

// Task #2
function checkSlices () {
    let a: number = 1;
    let b: number = 8;
    const isBetween = (a, b, i) => i > a && i < b ;
    return isBetween;
};

// Task #3
interface Toppings extends Pizza {
    get sauce(): any;
    set sauce(string: 'tomato' | 'alfredo' | 'bbq sauce')
    pineapple?: boolean;
    parmesan?: boolean;
    crust?: boolean;
};
```

PE07

PE08

```
class MultiplicatorUnitFailure extends Error {}

function primitiveMultiply(x, y) {
  if (Math.random() < 0.2) {
    return x * y;
  } else {
    throw new MultiplicatorUnitFailure("Break");
  }
}

function reliableMultiply(x, y) {
  "use strict";
  try {
    return primitiveMultiply(x, y);
  } catch (err) {
    if (err instanceof MultiplicatorUnitFailure) {
      return reliableMultiply(x, y);
    } else {
      throw err;
    }
  }
}

var i = Math.random().toFixed(2);
var j = Math.random().toFixed(2);
console.log(reliableMultiply(i, j));
```

```
function myFunction() {
  var uI = prompt("Please input a number between 1 & 20. ");
  if (typeof uI === "number" && uI >= 1 && uI <= 20) {
    console.log('"'.concat(uI, ' is a number between 1 & 20!"'));
    return true;
  }
  try {
    console.log('"'.concat(uI, ' that is not a valid answer"'));
    while (p1 > 0) {
      p2 = p1 * p2;
      p1 -= 1;
      return p1;
    }
  } finally {
    if (p1 <= 0) {
      return p2;
    }
    var x = 0;
    var j = 0;
    while (i > 0) {
      var alpha = x + y;
      x = y;
      y = alpha;
      j += 1;
    }
  }
}

myFunction();
```

Pirate Treasure The locked box

```
const box = {
    locked: true,
    unlock() { this.locked = false; },
    lock() { this.locked = true; },
    _content: [],
    get content() {
        if (this.locked) throw new Error("Locked!");
        return this._content;
    }
};

function withBoxUnlocked(body) {
    var lkd = box.locked;
    if (lkd) {
        box.unlock();
    } try {
        body();
    } finally {
        box.lock();
    }
}
console.log(box.locked); //true

withBoxUnlocked(function () {
    box.content.push("gold piece");
});
console.log(box._content); // accessing array

try {
    withBoxUnlocked(function () {
        throw new Error("Pirates on the horizon! Abort!");
    });
} catch (e) {
    console.log("Error raised:", e);
    console.log(box.locked);
} finally {
    box.unlock();
    withBoxUnlocked(function () {
        box.content.push("gems of arab`i");
    console.log(box.locked); // F
    });
}
```

PE09

```
var bigOak = require("./crow-tech").bigOak;
const anyStorage = require("./async.js").anyStorage;

async function locateScalpel(nest: any) {
  let stEnt = nest.name; //current location of the scalpel
  for (;;) {
    //recursive loop
    let next = await anyStorage(nest, stEnt, "scalpel");
    if (next == stEnt) return stEnt;
    else stEnt = nest;
  }
}

function woAsync(nest: any) {
  function rec(stEnt: any) {
    return anyStorage(nest, stEnt, "scalpel").then((next: any) => {
      if (next == stEnt) return stEnt;
      else return rec(next);
    });
  }
  return rec(nest.name);
}

locateScalpel(bigOak).then(console.log);
woAsync(bigOak).then(console.log);
```

promise_All

```
function Promise_all(p: any): Promise<any> {
  return new Promise((resolve, reject) => {
    let arr: any[] = [];
    let pl = p.length;
    for (let x = 0; x < pl; x++) {
      p[x].then((i: never) => {
        arr[x] = i;
        pl--;
        if (pl === 0) resolve(arr);
      }).catch(reject);
    } if (pl === 0) resolve(arr);
  });
}

// Test code.
```

```
Promise_all([]).then((array) => {
  console.log("This should be []:", array);
});
function soon(val: any) {
  return new Promise((resolve) => {
    setTimeout(() => resolve(val), Math.random() * 500);
  });
}
Promise_all([soon(1), soon(2), soon(3)]).then((array) => {
  console.log("This should be [1, 2, 3]:", array);
});
Promise_all([soon(1), Promise.reject("X"), soon(3)])
  .then((array) => {
    console.log("We should not get here");
  })
  .catch((error) => {
    if (error !== "X") {
      console.log("Unexpected failure:", error);
    }
  });
```

1.

```

```

PE10

```

```