For each of the prompts, write the answer to the prompt in Python on the left side, write the answer to the prompt in Javascript on the right side. The best way to do this is to write a small program in the appropriate IDE, then copy/paste the *tested and working* code into the appropriate box.

1. Declare a Variable with a value. Reassign a variable’s value

|  |  |
| --- | --- |
| amongus = 69  amongus = 420 | let amongus = 69;  amongus = 420; |

1. Assign the result of an expression to a variable.

|  |  |
| --- | --- |
| amongus = (9 + 10) | let amongus = (9 + 10); |

1. Change the data type of a variable’s value and assign it to the original value
   1. Convert to string
   2. Convert to number (integer or float)

|  |  |
| --- | --- |
| amongus = 10  amongus = str(amongus)  print(amongus)  print(type(amongus))  amongus = int(amongus)  amongus += 5  print(amongus)  print(type(amongus)) | let amongus;  amongus = 10  amongus = String(amongus)  console.log(amongus)  console.log(typeof amongus)  amongus = Number(amongus)  amongus += 5  console.log(amongus)  console.log(typeof amongus) |

1. Create a conditional statement that branches depending on if a variable’s value is greater than or equal to 7, greater than 4, or neither

|  |  |
| --- | --- |
| import random  amongus = random.randint(1, 10)  print(amongus)  if amongus > 4 and amongus < 7: (      print("Lol, mid")  )  elif amongus >= 7: (      print("High as a number higher than or equal to 7")  )  else: (      print("Wow unlucky, you suck lol")  ) | let amongus = Math.floor(Math.random()\*11)  console.log(amongus)  if (amongus > 4 && amongus < 7) {      console.log('Lol, mid')  }  else if (amongus >= 7) {      console.log('High as a number higher than or equal to 7')  }  else {      console.log('Wow unlucky, you suck lol')  } |

1. Create a conditional statement that only branches if the variable’s value is ‘blue’ OR ‘green’

|  |  |
| --- | --- |
| import random  colors = ['Red', 'Green', 'Blue', 'Purple', 'Orange', 'Pink']  color = random.choice(colors)  print(color)  if color == 'Blue' or color == 'Green': (      print("Im Grue Dabudi Dabudie")  )  else: (      print("If I was literally anything else I would die")  ) | import random  colors = ['Red', 'Green', 'Blue', 'Purple', 'Orange', 'Pink']  color = random.choice(colors)  print(color)  if color == 'Blue' or color == 'Green': (      print("Im Grue Dabudi Dabudie")  )  else: (      print("If I was literally anything else I would die")  ) |

1. Create a conditional statement that only branches if the variable’s value is ‘black’ AND ‘yellow’

|  |  |
| --- | --- |
| import random  colors = ['Red', 'Green', 'Blue', 'Purple', 'Yellow', 'Black']  color = random.sample(colors, 2)  print(color)  if 'Blue' in color or 'Green' in color: (      print("Im Grue Dabudi Dabudie")  )  elif 'Black' in color and 'Yellow' in color: (      print('Black and Yellow, Black and Yellow, Black and Yellow, Black and Yellow')  )  else: (      print('If I was literally anything else I would die')  ) | function getMultipleRandom(colors, num) {      const shuffled = [...colors].sort(() => 0.5 - Math.random());      return shuffled.slice(0, num);  }  let colors = [      'Red', 'Green', 'Blue', 'Purple', 'Yellow', 'Black'  ];  color = getMultipleRandom(colors, 2);  console.log(color);  console.log(color.indexOf('Blue'))  if (color.indexOf('Blue') != -1 || color.indexOf('Green') != -1) {      console.log("Im Grue Dabudi Dabudie")  }  else if (color.indexOf('Black') != -1 && color.indexOf('Yellow') != -1) {      console.log('Black and Yellow, Black and Yellow, Black and Yellow, Black and Yellow')  }  else {      console.log("If I was literally anything else I would die")  }; |

1. Create a function that outputs “Hello” to the console

|  |  |
| --- | --- |
| def bozo():      print('Hello')  bozo() | function bozo() {      console.log('Hello')  }  bozo() |

1. Create a function that returns “Hello”

|  |  |
| --- | --- |
| def bozo():      return 'Hello'  print(bozo()) | function bozo() {      return 'Hello'  }  console.log(bozo()) |

1. Create a function that takes two arguments, adds their values together, assigns the new value to a local variable, and returns that variable

|  |  |
| --- | --- |
| def bozo(x, y):      z = x + y      return z  among = 9  us = 10  print(bozo(among, us)) | function bozo(x, y) {      let z;      z = x + y      return z  }  let among = 9  let us = 10  console.log(bozo(among, us)) |

1. Create a loop that repeats while a variable’s value is true

|  |  |
| --- | --- |
| amongus = True  while amongus:      print('corona') | let amongus = true;  while (amongus) {      console.log('corona')  } |

1. Create a loop that repeats while a variable’s value is true, but will break mid-loop if another variable equals false

|  |  |
| --- | --- |
| import random  among = True  us = True  while among and us:      print('corona')      us = random.randint(1, 4)      if us == 1:          us = False | let among = true;  let us = 5;  while (among && us == 5) {      console.log('corona')      among = Math.floor(Math.random() \* 3)      if (us == 1) {          us = false      }  } |

1. Create an array/list that contains three strings

|  |  |
| --- | --- |
| amongus = ['The', 'Impostor', 'Sus']  print(amongus) | let amongus = ['The', 'Impostor', 'Sus']  console.log(amongus) |

1. Using the array from #12, remove the last element in the array

|  |  |
| --- | --- |
| amongus = ['The', 'Impostor', 'Sus']  print(amongus)  amongus.pop(2)  print(amongus) | let amongus = ['The', 'Impostor', 'Sus']  console.log(amongus)  amongus.pop(3)  console.log(amongus) |

1. Using the array from #12, remove the first element in the array

|  |  |
| --- | --- |
| amongus = ['The', 'Impostor', 'Sus']  print(amongus)  amongus.pop(0)  print(amongus) | let amongus = ['The', 'Impostor', 'Sus']  console.log(amongus)  amongus = amongus.slice(1)  console.log(amongus) |

1. Using the array from #12, remove the element in the middle (index 1)

|  |  |
| --- | --- |
| amongus = ['The', 'Impostor', 'Sus']  print(amongus)  del amongus[1]  print(amongus) | let among = ['The', 'Impostor', 'Sus'];  let it = 0;  let us;  function arr(x, y) {      x = y.pop()      y.pop()      y.push(x)      return y  };  console.log(arr(us, among)) |

1. Using the array from #12, output the first element

|  |  |
| --- | --- |
| amongus = ['The', 'Impostor', 'Sus']  print(amongus)  print(amongus[0]) | let amongus = ['The', 'Impostor', 'Sus'];  console.log(amongus[0]) |

1. Create an empty array. Then add a string element to it

|  |  |
| --- | --- |
| amongus = []  amongus.append('Sus')  print(amongus) | let amongus = [];  amongus.push("Sus")  console.log(amongus) |

1. Using the array from #12, use a loop to output each element

|  |  |
| --- | --- |
| amongus = ['The', 'Impostor', 'Sus']  it = 0  while it < 3:      print(amongus[it])      it += 1 | let amongus = ['The', 'Impostor', 'Sus'];  let it = 0;  while (it < 3) {      console.log(amongus[it])      it += 1  } |

1. Using the array from #12, use a loop to output the index of each element

|  |  |
| --- | --- |
| amongus = ['The', 'Impostor', 'Sus']  it = 0  while it < 3:      print(amongus.index(amongus[it]))      it += 1 | let amongus = ['The', 'Impostor', 'Sus'];  let it = 0;  while (it < 3) {      console.log(amongus.indexOf(amongus[it]))      it += 1  } |

1. Create an object with one property and one method. The property value will be a string, and the method will output the value of this object’s property.

|  |  |
| --- | --- |
| class Impostor:      def \_\_init\_\_(self, role):          self.role = 'Impostor'      def character(self):          print(self.role)  a = Impostor('Impostor')  print(type(a).\_\_name\_\_) | let amongus = {      role: "impostor",      character: function () {          return role      }  }; |

1. Using the object in #20, print the name of each property/method key in the object

|  |  |
| --- | --- |
| class Impostor:      def \_\_init\_\_(self):          self.role = 'Impostor'          self.color = 'Red'      def character(self):          print(self.role)  obj = Impostor()  obj\_vars = vars(obj)  print(obj\_vars) | const amongus = {      role: "Impostor",      character: function () {          return this.role      }  };  console.log(amongus.character()) |

1. Create a variable whose value is a string equal to the name (key) of the property of the object in #20. Output the property value of the object by using the variable (not the property name)

|  |  |
| --- | --- |
| class Impostor:      def \_\_init\_\_(self, role):          self.role = role      def character(self):          print(self.role)  amongus = Impostor("Impostor")  amongus.character() | function getKeyByValue(object, value) {      return Object.keys(object).find(key =>          object[key] === value);  }  let us = 'impostor'  const among = {      role: 'impostor',      character: function () {          return this.role      }  };  console.log(getKeyByValue(among, us)) |

1. Create a two-dimensional array
   1. Create an array/list with three elements
   2. Each element is another array/list with three elements whose values are strings

|  |  |
| --- | --- |
| amongus = [      ['A', 'Word', 'lol']      ['Status', 'Of', 'Mogus']      ['Actions', 'Be', 'Impostorlike']  ] | let amongus = [      ['A', 'Word', 'lol']      ['Status', 'Of', 'Mogus']      ['Actions', 'Be', 'Impostorlike']  ]; |

1. Using the array in #23, use two loops (one inside the other) to output all 9 elements in the multi-dimensional array.

|  |  |
| --- | --- |
| amongus = [      ['A', 'Word', 'lol'],      ['Status', 'Of', 'Mogus'],      ['Actions', 'Be', 'Impostorlike']  ]  for items in range(len(amongus)):      for y in range (len(amongus[items])):          print(amongus[items][y]) | let amongus = [      ['A', 'Word', 'lol'],      ['Status', 'Of', 'Mogus'],      ['Actions', 'Be', 'Impostorlike']  ];  for (var x = 0; x < amongus.length; x++) {      var amonguses = amongus[x];      for (var y = 0; y < amonguses.length; y++) {          console.log("amonguses[" + x + "][" + y + "] = " + amonguses[y]);      }  } |

1. Add a new property to the object used in #20 with a number value

|  |  |
| --- | --- |
| class Impostor():      def \_\_init\_\_(self, role):          self.role = 'Impostor'      def character(self):          print(self.role)  unalives = Impostor('red')  setattr(unalives, 'salary', 0)  print(getattr(unalives, 'salary')) | let amongus = {      role: "impostor",      character: function () {          return role      }  };  amongus.unalives = 0; |