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

|  |  |
| --- | --- |
| X = ‘TONY’ | Var X = ‘BOB’ |

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

|  |  |
| --- | --- |
| X = 5 + 6 | Var X = 10586985 +4455478 |

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)

|  |  |
| --- | --- |
| Int(7 + false) | Boltonum = 50 + true |

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

|  |  |
| --- | --- |
| If x = 7: (print“hub”) else if : x=4 (print“jam”) else: (print“sob”) | If(time === 7){}  Else if (time===4){}  Else{} |

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

|  |  |
| --- | --- |
| If X==”blue” or x== “green” | If (x == “blue” || x == “green”){} |

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

|  |  |  |
| --- | --- | --- |
| If X==”black” & x== “yellow” | If (x == “black” & x == “yellow”){} |  |

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

|  |  |
| --- | --- |
| Print(“hello”) | Console.log(“hello”) |

1. Create a function that returns “Hello”

|  |  |
| --- | --- |
| Def say ()  Print(“hello”)  Return | Function say()  Console.log(“hello”)  return |

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 add (x , y)  x+y=  Return value | Function add(x , y)  X + y  Retun value |

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

|  |  |
| --- | --- |
| While x =true : x = true | While (x = true){var x = true } |

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

|  |  |
| --- | --- |
| I=0  While true:  Print (i)  I = 5  If I == 5:  break | Var X = true  While (x = true){var x = true } |

1. Create an array/list that contains three strings

|  |  |
| --- | --- |
| From array import \*  Cats = array(‘I’ ,[“nugget”,”ivy”,”miso”])  Print(cats | Const cats = [“nugget” , “ivy”,”miso”]  Consol.log(cats) |

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

|  |  |
| --- | --- |
| From array import \*  Cats = array(‘I’ ,[“nugget”,”ivy”,”miso”])  Cats.remove(2)  Print(cats) | Const cats = [“nugget” , “ivy”,”miso”]  Delete cats[2]  Consol.log(cats) |

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

|  |  |
| --- | --- |
| From array import \*  Cats = array(‘I’ ,[“nugget”,”ivy”,”miso”])  Cats.remove(0)  Print(cats) | Const cats = [“nugget” , “ivy”,”miso”]  Delete cats[0]  Consol.log(cats) |

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

|  |  |
| --- | --- |
| From array import \*  Cats = array(‘I’ ,[“nugget”,”ivy”,”miso”])  Cats.remove(1)  Print(cats) | Const cats = [“nugget” , “ivy”,”miso”]  Delete cats[1]  Consol.log(cats) |

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

|  |  |
| --- | --- |
| From array import \*  Cats = array(‘I’ ,[“nugget”,”ivy”,”miso”])  First = [cats[0]]  Print(first) | Const cats = [“nugget” , “ivy”,”miso”]  Const first = cats[0]  Consol.log(first) |

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

|  |  |
| --- | --- |
| From array import \*  shopping = array(‘I’[])  shopping.append(“BackDoor”)  Print(shopping) | Const shopping =[]  shoppping.push(“MicroChipedXlDogDoor”)  Consol.log(shopping) |

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

|  |  |
| --- | --- |
| From array import \*  Cats = array(‘I’ ,[“nugget”,”ivy”,”miso”])  For pet in range(0,len(cats));  Print(cats[pet]) | Const cats = [“nugget” , “ivy”,”miso”]  Cats.forEach((pet) => cosole.log(pet)); |

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

|  |  |  |
| --- | --- | --- |
| From array import \*  Cats = array(‘I’ ,[“nugget”,”ivy”,”miso”])  For index ,val in enumerate(cats):  Print (index,val) | Const cats = [“nugget” , “ivy”,”miso”]  Cats.forEach((pet , index) => cosole.log(pet , index)); |  |

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 picnic:  Def sandwitch(subway,type):  Subway.type = type  AntiBurnOut=picnic(“coldCut”)  printt (AntiBurnOut.type) |  |

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

|  |  |
| --- | --- |
|  |  |

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)

|  |  |
| --- | --- |
|  |  |

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

|  |  |
| --- | --- |
|  |  |

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

|  |  |
| --- | --- |
|  |  |

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

|  |  |
| --- | --- |
|  |  |