|  |  |  |
| --- | --- | --- |
| Assign a value to a variable:  variable\_name = value | age = 16 | Variables store information for later use. |
| Define a function:  def functionName(**argument\_name**):  #Body  Call a function: functionName(argument\_value) | def printHomeland(**country**):  print("I am from "+  country)  printHomeland("Brazil") | Functions are bite-sized actions that we write to use, reuse, and keep our code organized. |
| A chain of if, elif, and else statements:  if condition:    #Body elif condition:    #Body elif condition:    #Body else:    #Body | a = 200 b = 33 if b > a:    print("b is greater ") elif a == b:    print("they are equal") else:    print("a is greater ") | if, elif, and else are used to change what code runs based on different conditions.  If-statements are detours through the code. Code “inside” an if, elif, or else must be indented. elif and else must follow an if or elif. |
| Create a while loop:  while condition:    #Body | #Print Interesting! until  #user says quit  response = input("What?") while response != "quit":    print("Interesting!")    response=input("What?") | While loops repeat as long as their condition is true. While loops are used when you don’t know how long a loop will repeat. |
| Create a loop to repeat a specific number of times:  for var\_name in range(number):   #Body | #Print the numbers 0  #through 5  for i in range(6):   print(i) | This loop is used when you know exactly how many times you want to repeat a loop. |
| Create a loop to access each value in a list:  for var\_name in list:   #Body | #Print each fruit in a  #fruit list  fruits=["apple","pear","grape"]  for x in fruits:    print(x) | This loop is used for “walking” through a list of values. |
| Create an empty list and add two values to it:  my\_list = []  my\_list.append(value)  my\_list.append(other\_value) | #Create a fruit list  fruits=["apple","pear","grape"]  #Add orange to end of list  fruits.append("orange")  #Print list length: 4  print(len(fruits))  #Delete first item in list  del fruits[0]  #Print list:pear,grape,orange print(fruits) | Whenever you are tempted to create numbered variables like: noun1, noun2, noun3, you should create a list instead. |
| Create a class with one function:  class ClassName:   def \_\_init\_\_(self,parameters):      self.variable\_name = value    def functionName(self):      #Body  Create an instance of the class:  var\_name = ClassName(parameters) | #Create a Person class  class Person:   def \_\_init\_\_(self,name,age):     self.name = name     self.age = age    def greet(self):     print("Hello my name is "+  self.name)  #Create an instance of Person john = Person("John", 36)  #Call this person’s greet  #function  john.greet() | Classes define objects. Objects are collections of variables and functions that work together. We create objects to organize our code. |