# Exercise 1.1 – Variable Names

The Python interpreter has strict rules for variable names. Which of the following are legal Python names? If the name is not legal, state the reason.  
  
  
1. and  
Not legal: and is a python keyword

2. \_and  
Legal

3. var  
Not legal: var is a python keyword

4. 1var  
Legal  
  
  
  
  
5. COLOR

Legal: uppercase used for constants usually

# Exercise 1.2 – Types

It is important that we know the type of values stored in a variable so that we can use the correct operators. Python automatically infers the type from the value you assign to the variable. Write down the types of values stored in each of the variables below. Pay special attention to punctuation: values are not always what they seem!

1. a = False  
Boolean

2. b = 3.7

Float

3. c = ‘Alex’

string

4. d = ‘17’

string

5. e = 17  
  
integer

# Exercise 1.3 – Boolean operators

Boolean operators can seem tricky at first, and it takes practice to evaluate them correctly. Write the value (True or False) produced by each expression below, using the assigned values of the variables a, b, and c. **Hint:** Work from the inside out, starting with the inner-most expressions, like in arithmetic.

a = False

b = True

c = False

1. b and c  
   False
2. b or c  
   True
3. not a and b

True

# Exercise 1.4 – Conditionals

The purpose of this exercise is to understand conditionals. Jimmy is looking for his dream job but has some restrictions. He loves Newfoundland and would take a job there if it paid over $40,000 a year. He hates Ontario and demands at least $100,000 to work there. Any other places he’s content to work for $60,000 a year unless he can work in space in which he would work there for free. The following code shows his basic strategy for evaluating a job offer.

|  |
| --- |
| **pay = \_\_\_\_\_**  **location = \_\_\_\_\_**  **if location == "Mars":"**  **print "So long, suckers! I’ll take it!"**  **elif location == "Ontario":**  **if pay < 100000:**  **print "No way"**  **else:**  **print "I’ll take it!"**  **elif location == "Newfoundland" and pay > 40000:**  **print "I’ll take it!"**  **elif pay > 60000:**  **print "I’ll take it!"**  **else:**  **print** **"No thanks, I can find something better."** |
|  |

For each of the following job offers, write out the output that would be generated. It is important to be able to understand what a piece of code does without running it.

1. location = “Ontario”  
    pay = 50000  
     
   “No Way”
2. location = “Quebec”  
    pay = 50000  
   "No thanks, I can find something better."
3. location = “Mars”  
    pay = 5  
   “So long, suckers! I’ll take it!”

# Exercise 1.5 – Understanding Loops

For each of the following code snippers, write what the output would be, without running the code. (EACH COMMA IS A NEW LINE)  
  
1. num = 10  
 while num < 3:  
 print num  
 num = num – 1

Nothing will output  
  
  
  
  
  
2. divisor = 2  
 for i in range (0, 10, 2):  
 print (i/divisor)

0.0, 1.0, 2.0, 3.0, 4.0

3. num = 10  
 while True:  
 if num < 7:   
 break  
 print (num)  
 num -= 1

10, 9, 8, 7

# Exercise 1.6 – Understanding Functions Which of the following code snippets correctly defines a Python function that takes two argument, divides them and returns a whole number?

A) def add(a, b):

return a \* b

B) def add(a, b):

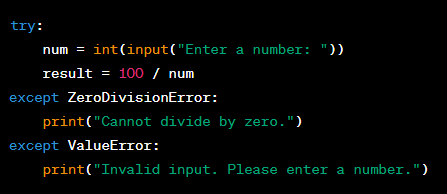
return a + b

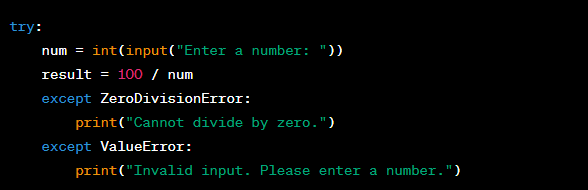
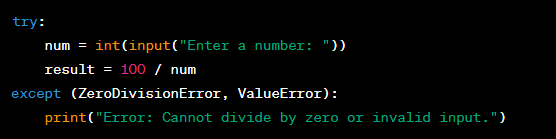
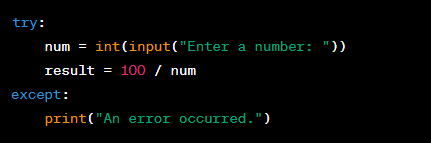
C) def add(a, b):

return a / b

D) def add(a, b):

return a // b

Exercise 1.7 – Understanding Exception Handling  
  
Which of the following code snippets demonstrates the correct usage of exception handling in Python?  
  
A)  
 

B)   
  
  
  
  
C)  
   
  
  
  
  
D)  
 

Exercise 1.8 – Understanding Nested Lists  
  
Refer to the below code:

  
  
Which of the following options correctly accesses the value 8 from the ‘nested\_list’?

A) nested\_list[2][1][2]

B) nested\_list[2][1]

C) nested\_list[3][0]

D) nested\_list[2][1][1]