

Introduction to Machine Learning with Python

Maths operations, functions, if blocks

Dr. Süha Tuna
İTÜ Informatics Institute

- Math operators, random numbers
- Functions, `if`, `elif`
- Counting
- Summing
- Flags
- Swapping

Math operators, order of operations

Operator	Description
+	addition
-	subtraction
*	multiplication
/	division
**	exponentiation
//	integer division
%	modulo (remainder)

Please **E**xcuse **M**e **D**ear **A**unt **S**ally

P	() Parentheses
E	A^2 Exponents
M	\times Multiplication
	or
D	\div Division
A	+ Addition
	or
S	- Subtraction

Random numbers

```
from random import randint
```

```
from random import randint  
x = randint(1,10)  
print('A random number between 1 and 10: ', x)
```

```
A random number between 1 and 10: 7
```

Math functions

```
from math import sin, pi
print('Pi is roughly', pi)
print('sin(0) =', sin(0))
```

```
Pi is roughly 3.14159265359
sin(0) = 0.0
```

```
print(abs(-4.3))
print(round(3.336, 2))
print(round(345.2, -1))
```

```
4.3
3.34
350.0
```

Getting help from Python

```
>>> import math  
>>> dir(math)
```

```
['__doc__', '__name__', '__package__', 'acos', 'acosh', 'asin',  
'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos',  
'cosh', 'degrees', 'e', 'exp', 'fabs', 'factorial', 'floor',  
'fmod', 'frexp', 'fsum', 'hypot', 'isinf', 'isnan', 'ldexp',  
'log', 'log10', 'log1p', 'modf', 'pi', 'pow', 'radians', 'sin',  
'sinh', 'sqrt', 'tan', 'tanh', 'trunc']
```

Exercises

1. Write a program that generates and prints 50 random integers, each between 3 and 6.
2. Write a program that generates a random number, x , between 1 and 50, a random number y between 2 and 5, and computes x^y .
3. Write a program that generates a random number between 1 and 10 and prints your name that many times.
4. Write a program that generates a random decimal number between 1 and 10 with two decimal places of accuracy. Examples are 1.23, 3.45, 9.80, and 5.00.
5. Write a program that generates 50 random numbers such that the first number is between 1 and 2, the second is between 1 and 3, the third is between 1 and 4, ..., and the last is between 1 and 51.
6. Write a program that asks the user to enter two numbers, x and y , and computes $\frac{|x-y|}{x+y}$.
7. Write a program that asks the user to enter an angle between -180° and 180° . Using an expression with the modulo operator, convert the angle to its equivalent between 0° and 360° .

Exercises

8. Write a program that asks the user for a number of seconds and prints out how many minutes and seconds that is. For instance, 200 seconds is 3 minutes and 20 seconds. [Hint: Use the // operator to get minutes and the % operator to get seconds.]
9. Write a program that asks the user for an hour between 1 and 12 and for how many hours in the future they want to go. Print out what the hour will be that many hours into the future. An example is shown below.

```
Enter hour: 8
How many hours ahead? 5
New hour: 1 o'clock
```

10.
 - (a) One way to find out the last digit of a number is to mod the number by 10. Write a program that asks the user to enter a power. Then find the last digit of 2 raised to that power.
 - (b) One way to find out the last two digits of a number is to mod the number by 100. Write a program that asks the user to enter a power. Then find the last two digits of 2 raised to that power.
 - (c) Write a program that asks the user to enter a power and how many digits they want. Find the last that many digits of 2 raised to the power the user entered.
11. Write a program that asks the user to enter a weight in kilograms. The program should convert it to pounds, printing the answer rounded to the nearest tenth of a pound.
12. Write a program that asks the user for a number and prints out the factorial of that number.
13. Write a program that asks the user for a number and then prints out the sine, cosine, and tangent of that number.

if statements

```
from random import randint

num = randint(1,10)
guess = eval(input('Enter your guess: '))
if guess==num:
    print('You got it!')
```

```
if guess==num:
    print('You got it!')
else:
    print('Sorry. The number is ', num)
```

Conditional operators

Expression	Description
<code>if x>3:</code>	if x is greater than 3
<code>if x>=3:</code>	if x is greater than or equal to 3
<code>if x==3:</code>	if x is 3
<code>if x!=3:</code>	if x is not 3

```
if grade>=80 and grade<90:  
    print('Your grade is a B.')  
if score>1000 or time>20:  
    print('Game over.')  
if not (score>1000 or time>20):  
    print('Game continues.')
```

```
if (score<1000 or time>20) and turns_remaining==0:  
    print('Game over.')
```

Common mistakes

Incorrect	Correct
<code>if x=1:</code>	<code>if x==1:</code>

```
if x>1 and x<100:  
if x>1 or x<100:
```

```
if grade>=80 and <90:
```

This will lead to a syntax error. We have to be explicit. The correct statement is

```
if grade>=80 and grade<90:
```

On the other hand, there is a nice shortcut that does work in Python (though not in many other programming languages):

```
if 80<=grade<90:
```

elif

```
grade = eval(input('Enter your score: '))

if grade >= 90:
    print('A')
if grade >= 80 and grade < 90:
    print('B')
if grade >= 70 and grade < 80:
    print('C')
if grade >= 60 and grade < 70:
    print('D')
if grade < 60:
    print('F')
```

```
grade = eval(input('Enter your score: '))

if grade >= 90:
    print('A')
elif grade >= 80:
    print('B')
elif grade >= 70:
    print('C')
elif grade >= 60:
    print('D')
else:
    print('F')
```

Exercises

1. Write a program that asks the user to enter a length in centimeters. If the user enters a negative length, the program should tell the user that the entry is invalid. Otherwise, the program should convert the length to inches and print out the result. There are 2.54 centimeters in an inch.
2. Ask the user for a temperature. Then ask them what units, Celsius or Fahrenheit, the temperature is in. Your program should convert the temperature to the other unit. The conversions are $F = \frac{9}{5}C + 32$ and $C = \frac{5}{9}(F - 32)$.
3. Ask the user to enter a temperature in Celsius. The program should print a message based on the temperature:
 - If the temperature is less than -273.15, print that the temperature is invalid because it is below absolute zero.
 - If it is exactly -273.15, print that the temperature is absolute 0.
 - If the temperature is between -273.15 and 0, print that the temperature is below freezing.
 - If it is 0, print that the temperature is at the freezing point.
 - If it is between 0 and 100, print that the temperature is in the normal range.
 - If it is 100, print that the temperature is at the boiling point.
 - If it is above 100, print that the temperature is above the boiling point.
4. Write a program that asks the user how many credits they have taken. If they have taken 23 or less, print that the student is a freshman. If they have taken between 24 and 53, print that they are a sophomore. The range for juniors is 54 to 83, and for seniors it is 84 and over.

Exercises

5. Generate a random number between 1 and 10. Ask the user to guess the number and print a message based on whether they get it right or not.
6. A store charges \$12 per item if you buy less than 10 items. If you buy between 10 and 99 items, the cost is \$10 per item. If you buy 100 or more items, the cost is \$7 per item. Write a program that asks the user how many items they are buying and prints the total cost.
7. Write a program that asks the user for two numbers and prints `Close` if the numbers are within `.001` of each other and `Not close` otherwise.
8. A year is a leap year if it is divisible by 4, except that years divisible by 100 are not leap years unless they are also divisible by 400. Write a program that asks the user for a year and prints out whether it is a leap year or not.
9. Write a program that asks the user to enter a number and prints out all the divisors of that number. [Hint: the `%` operator is used to tell if a number is divisible by something. See [Section 3.2](#).]
10. Write a multiplication game program for kids. The program should give the player ten randomly generated multiplication questions to do. After each, the program should tell them whether they got it right or wrong and what the correct answer is.

```
Question 1: 3 x 4 = 12
Right!
Question 2: 8 x 6 = 44
Wrong. The answer is 48.
...
...
Question 10: 7 x 7 = 49
Right.
```

Counting

Counting is an extremely common thing. The two things involved are:

1. `count=0` — Start the count at 0.
2. `count=count+1` — Increase the count by 1.

```
count = 0
for i in range(10):
    num = eval(input('Enter a number: '))
    if num>10:
        count=count+1
print('There are', count, 'numbers greater than 10.')
```

Counting

```
count1 = 0
count2 = 0
for i in range(10):
    num = eval(input('Enter a number: '))
    if num>10:
        count1=count1+1
    if num==0:
        count2=count2+1
print('There are', count1, 'numbers greater than 10.')
print('There are', count2, 'zeroes.')
```

```
count = 0
for i in range(1,101):
    if (i**2)%10==4:
        count = count + 1
print(count)
```


Summing

Example 1 This program will add up the numbers from 1 to 100. The way this works is that each time we encounter a new number, we add it to our running total, *s*.

```
s = 0
for i in range(1,101):
    s = s + i
print('The sum is', s)
```

Example 2 This program that will ask the user for 10 numbers and then computes their average.

```
s = 0
for i in range(10):
    num = eval(input('Enter a number: '))
    s = s + num
print('The average is', s/10)
```

Example 3 A common use for summing is keeping score in a game. Near the beginning of the game we would set the score variable equal to 0. Then when we want to add to the score we would do something like below:

```
score = score + 10
```

Swapping

Incorrect

```
x = y  
y = x
```

Correct

```
hold = x  
x = y  
y = hold
```

Python shorthand

```
x, y = y, x
```

Flag variables

A flag variable can be used to let one part of your program know when something happens in another part of the program. Here is an example that determines if a number is prime.

```
num = eval(input('Enter number: '))

flag = 0
for i in range(2, num):
    if num%i==0:
        flag = 1

if flag==1:
    print('Not prime')
else:
    print('Prime')
```

Maxes and mins

A common programming task is to find the largest or smallest value in a series of values. Here is an example where we ask the user to enter ten positive numbers and then we print the largest one.

```
largest = eval(input('Enter a positive number: '))
for i in range(9):
    num = eval(input('Enter a positive number: '))
    if num > largest:
        largest = num
print('Largest number:', largest)
```

Comments

Single-line comments For a single-line comment, use the `#` character.

```
# a slightly sneaky way to get two values at once  
num1, num2 = eval(input('Enter two numbers separated by commas: '))
```

You can also put comments at the end of a line:

```
count = count + 2 # each divisor contributes two the count
```

Multi-line comments For comments that span several lines, you can use triple quotes.

```
""" Program name: Hello world  
    Author: Brian Heinold  
    Date: 1/9/11 """  
  
print('Hello world')
```

Exercises

1. Write a program that counts how many of the squares of the numbers from 1 to 100 end in a 1.
2. Write a program that counts how many of the squares of the numbers from 1 to 100 end in a 4 and how many end in a 9.
3. Write a program that asks the user to enter a value n , and then computes $(1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}) - \ln(n)$. The \ln function is `log` in the `math` module.
4. Write a program to compute the sum $1 - 2 + 3 - 4 + \dots + 1999 - 2000$.

Exercises

5. Write a program that asks the user to enter a number and prints the sum of the divisors of that number. The sum of the divisors of a number is an important function in number theory.
6. A number is called a *perfect number* if it is equal to the sum of all of its divisors, not including the number itself. For instance, 6 is a perfect number because the divisors of 6 are 1, 2, 3, 6 and $6 = 1 + 2 + 3$. As another example, 28 is a perfect number because its divisors are 1, 2, 4, 7, 14, 28 and $28 = 1 + 2 + 4 + 7 + 14$. However, 15 is not a perfect number because its divisors are 1, 3, 5, 15 and $15 \neq 1 + 3 + 5$. Write a program that finds all four of the perfect numbers that are less than 10000.
7. An integer is called *squarefree* if it is not divisible by any perfect squares other than 1. For instance, 42 is squarefree because its divisors are 1, 2, 3, 6, 7, 21, and 42, and none of those numbers (except 1) is a perfect square. On the other hand, 45 is not squarefree because it is divisible by 9, which is a perfect square. Write a program that asks the user for an integer and tells them if it is squarefree or not.
8. Write a program that swaps the values of three variables x , y , and z , so that x gets the value of y , y gets the value of z , and z gets the value of x .
9. Write a program to count how many integers from 1 to 1000 are not perfect squares, perfect cubes, or perfect fifth powers.

Exercises

10. Ask the user to enter 10 test scores. Write a program to do the following:
 - (a) Print out the highest and lowest scores.
 - (b) Print out the average of the scores.
 - (c) Print out the second largest score.
 - (d) If any of the scores is greater than 100, then after all the scores have been entered, print a message warning the user that a value over 100 has been entered.
 - (e) Drop the two lowest scores and print out the average of the rest of them.
11. Write a program that computes the factorial of a number. The factorial, $n!$, of a number n is the product of all the integers between 1 and n , including n . For instance, $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$. [Hint: Try using a multiplicative equivalent of the summing technique.]
12. Write a program that asks the user to guess a random number between 1 and 10. If they guess right, they get 10 points added to their score, and they lose 1 point for an incorrect guess. Give the user five numbers to guess and print their score after all the guessing is done.
13. In the last chapter there was an exercise that asked you to create a multiplication game for kids. Improve your program from that exercise to keep track of the number of right and wrong answers. At the end of the program, print a message that varies depending on how many questions the player got right.

Exercises

Exercises