



AUSTRALIAN MATHS TRUST

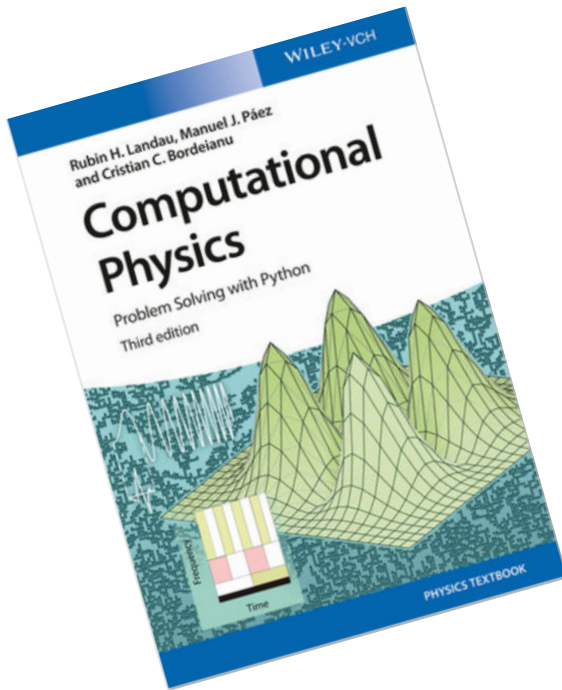
Programming in Python

Workbook



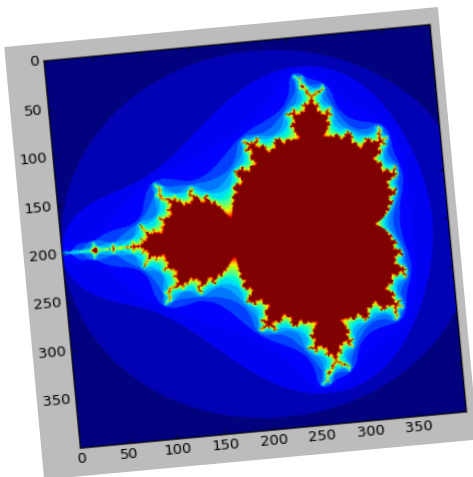
Programming in Python

An easy-to-read language, on the way to C++,
used in all sorts of sciences,
used with Django to design websites,
...

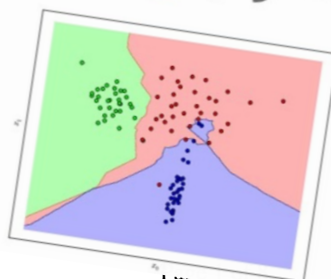
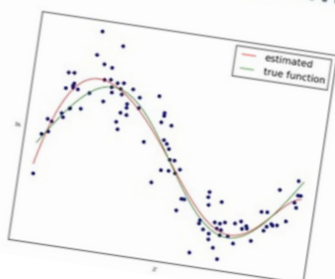


```
>>> for n in range(6):  
    print(2**n)  
  
1  
2  
4  
8  
16  
32
```

```
>>> n = 0  
>>> while n < 6:  
    print(2**n)  
    n = n + 1  
  
1  
2  
4  
8  
16  
32
```




Machine Learning with  python



Learning Python on www.groklearning.com

Courses, ready when you are!

MODULES
8




PROBLEMS
32

Introduction to Programming

An introductory course using the programming language Python for people with no programming experience.

[Start](#)[Details](#)

MODULES
8



PROBLEMS
32

Introduction to Programming 2

Finished Introduction to Programming and want more?

[Start](#)[Details](#)




Introduction to Programming 3

Mastered Python basics and want to try your hand at some more advanced programming?

[Coming soon!](#)

MODULES
1



PROBLEMS
6

Image Manipulation 1 in Python


Extend your Python skills to automatically edit and manipulate your images.

[Start](#)[Details](#)

Competitions, five weeks of challenge and learning, with leaderboards and certificates 😊

Mon 4 March – Sun 14 April:


Web.Comp 2019
HTML5 & CSS3
School-only
[More details](#)



Beginners | Intermediate

TBA, July – September:


Challenge
Python 3 & Blockly
School-only
[More details](#)



Learn to code as you compete!

TBA, October – December:

Code Quest
Python 3 & Blockly
School-only
[More details](#)



Go on a coding adventure.

Python 3 Cheatsheet

Interact with the user (input and output)

Print a message

```
print('Hello, world!')
```

Print multiple values (of different types)

```
ndays = 365
print('There are', ndays, 'in a year')
```

Asking the user for a string

```
name = input('What is your name? ')
```

Asking the user for a whole number (an integer)

```
num = int(input('Enter a number: '))
```

Text (strings)

Single quoted

```
'perfect'
```

Double quoted

```
"credit"
```

Multi-line

```
'''Hello,
World!'''
```

Add (concatenate) strings

```
'Hello' + 'World'
```

Multiply string by integer

```
'Echo...' * 4
```

Length of a string

```
len('Hello')
```

Convert string to integer

```
int('365')
```

Variables

Creating a variable

```
celsius = 25
```

Using a variable

```
celsius*9/5 + 32
```

Whole numbers (integers)

Addition and subtraction

```
365 + 1 - 2
```

Multiplication and division

```
25*9/5 + 32
```

Powers (2 to the power of 8)

```
2**8
```

Convert integer to string

```
str(365)
```

Decide between options

Decide to run a block (or not)

```
x = 3
if x == 3:
    print('x is 3')
```

Decide between two blocks

```
mark = 80
if mark >= 50:
    print('pass')
else:
    print('fail')
```

Decide between many blocks

```
mark = 80
if mark >= 65:
    print('credit')
elif mark >= 50:
    print('pass')
else:
    print('fail')
```

Are two values equal?

```
x == 3
```

Are two values not equal?

```
x != 3
```

Less than another?

```
x < 3
```

Greater than another?

```
x > 3
```

Less than or equal to?

```
x <= 3
```

Greater than or equal to?

```
x >= 3
```

△ two equals signs, not one

△ range starts from 0 and goes up to, but not including, 10

elif can be used without else

elif can be used many times

The answer is a Boolean:

```
True
```

or

```
False
```

String manipulation

Compare two strings

```
msg = 'hello'
if msg == 'hello':
    print('howdy')
```

Less than another string?

```
if msg < 'n':
    print('a-m')
else:
    print('n-z')
```

Convert to uppercase

```
msg.upper()
```

also lower and title

Count a character in a string

```
msg.count('l')
```

Replace a character or string

```
msg.replace('l', 'X')
```

Delete a character or string

```
msg.replace('l', '')
```

Is a character in a string?

```
'e' in msg
```

Is a string in another string?

```
'ell' in msg
```

△ strings are compared character at a time (lexicographic order)

also isupper and istitle

Repeat a block (a fixed number of times)

Repeat a block 10 times

```
for i in range(10):
    print(i)
```

Sum the numbers 0 to 9

```
total = 0
for i in range(10):
    total = total + i
print(total)
```

Repeat a block over a string

```
for c in 'Hello':
    print(c)
```

Keep printing on one line

```
for c in 'Hello':
    print(c, end=' ')
print('!!')
```

Repeat a block over list (or string) indices

```
msg = 'I grok Python!'
for i in range(len(msg)):
    print(i, msg[i])
```

Count from 0 to 9

```
range(10)
```

Count from 1 to 10

```
range(1, 11)
```

Count from 10 down to 1

```
range(10, 0, -1)
```

Count 2 at a time to 10

```
range(0, 11, 2)
```

Count down 2 at a time

```
range(10, 0, -2)
```

Putting it together: Celsius to Fahrenheit converter

Ask the user for a temperature in degrees Celsius

```
celsius = int(input('Temp. in Celsius: '))
```

Calculate the conversion

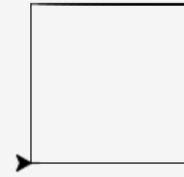
```
fahrenheit = celsius*9/5 + 32
```

Output the result

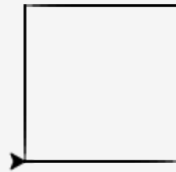
```
print(fahrenheit, 'Fahrenheit')
```

Python turtle activity on www.trinket.io

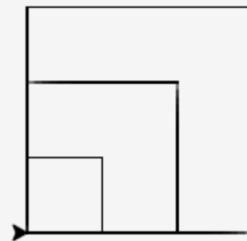
```
1 import turtle
2 jan = turtle.Turtle()
3
4 jan.forward(100)
5 jan.left(90)
6 jan.forward(100)
7 jan.left(90)
8 jan.forward(100)
9 jan.left(90)
10 jan.forward(100)
11 jan.left(90)
```



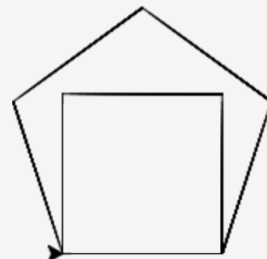
```
1 import turtle
2 jan = turtle.Turtle()
3
4 for i in range(4):
5     jan.forward(100)
6     jan.left(90)
```



```
1 import turtle
2 jan = turtle.Turtle()
3
4 def square(length):
5     for i in range(4):
6         jan.forward(length)
7         jan.left(90)
8
9 square(50)
10 square(100)
11 square(150)
```



```
1 import turtle
2 jan = turtle.Turtle()
3
4 def poly(sides, length):
5     for i in range(sides):
6         jan.forward(length)
7         jan.left(360.0/sides)
8
9 poly(4, 100)
10 poly(5, 100)
```



```
1 import turtle
2 jan = turtle.Turtle()
3
4 def poly(sides, length):
5     for i in range(sides):
6         jan.forward(length)
7         jan.left(360.0/sides)
8
9 for j in range(3,15):
10     poly(j, 20)
```



```
1 import turtle
2 jan = turtle.Turtle()
3
4 def poly(sides, length):
5     for i in range(sides):
6         jan.forward(length)
7         jan.left(360.0/sides)
8
9 for j in range(15,2,-1):
10     if j % 2 == 1:
11         jan.color("red")
12     else:
13         jan.color("white")
14     jan.begin_fill()
15     poly(j, 30)
16     jan.end_fill()
```



Programming challenges on www.dmoj.ca



1. Register, and affiliate yourself with the **digIT Summer School** in the process
2. Click on *PROBLEMS*
3. Enjoy the smorgasbord of problems and climb the leaderboard!

The problems from the Canadian Computing Competition (CCC) are a good place to start if you would like to enter the world of competitive programming. The Junior Division always starts with a gentle problem as illustrated below.

Problem, CCC '04 J1 – Squares:

Gigi likes to play with squares. She has a collection of equal-sized square tiles. Gigi wants to arrange some or all of her tiles on a table to form a solid square. What is the side length of the largest possible square that Gigi can build?

For example, when Gigi has 9 tiles she can use them all to build a square whose side length is 3. But when she has only 8 tiles, the largest square that she can build has side length 2.

Write a program that inputs the number of tiles and then prints out the maximum side length. You may assume that the number of tiles is less than ten thousand.

Sample Input 1

9

Sample Output 1

The largest square has side length 3.

Sample Input 2

8

Sample Output 2

The largest square has side length 2.

Sample Input 3

7535

Sample Output 3

The largest square has side length 86.

Suggested solution:

```
# Read the input from the screen and store it as an integer.
number_of_tiles = int(input())

# Calculate the square root and round DOWN to the nearest integer.
side_length = int(number_of_tiles**(1/2))

# Print the answer on the screen.
print("The largest square has side length " + str(side_length) + ".")
```


Informatics training on <http://orac.amt.edu.au>

1. Click on *Training Site*
2. Click on *New User* to register
3. Enjoy hundreds of informatics problems 😊

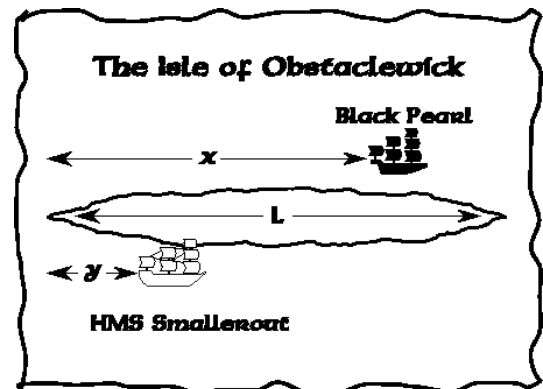
You will need to be able to read from a text file and write to a text file in order for you to be able to submit your solutions. Luckily that is quite easy as illustrated in the solution below.

Problem, AIO 2011 (Intermediate) – Pirates:

Yarr! Welcome aboard the Black Pearl! I'm Captain Mia Swamp, and this is my First Matey, Growlybills. We've heard you're handy with these computing contraptions, so I'll make you a deal: help us out with a little problem, and we won't feed you to the sharks.

You're in? Thought so.

See that map yonder? That long, thin island there is the Isle of Obstaclewick. Boring place. All you need to know is that it's L nautical miles long from east to west, and so thin we all just say it has zero width.



Our ship, the Black Pearl, is sailing the north coast of the island, X nautical miles from the west point. See the other ship, the one sailing the south coast, Y nautical miles from the west point? That's the HMS Smallerout, our target. It may look like a wibbly-wobbly old thing, but it's carrying some of Britain's greatest treasures.

We can sail either way around the Isle of Obstaclewick, approaching the Smallerout from either side. What we want you to do is tell us which way is shorter. We don't want to overwork the... volunteers... in the galley. So that's your job, landlubber! Write us a program that calculates the shortest distance we have to sail to reach the Smallerout!

| | | |
|-------------------------------|----------------------------------|-------------------------------|
| Sample Input 1 6 4 1 | Sample Input 2 10 10 10 | Sample Input 3 9 2 7 |
| Sample Output 1 5 | Sample Output 2 0 | Sample Output 3 9 |

Suggested solution:

```
# Read the lines of data from the file piratein.txt
lines = open("piratein.txt", "r").readlines()

# save the three variables as integers.
L = int(lines[0])
X = int(lines[1])
Y = int(lines[2])

# Calculate the two distances and pick the shortest!
one_way = X + Y
other_way = (L-X) + (L-Y)
answer = min(one_way, other_way)

# Write the answer to the file pirateout.txt
outfile = open("pirateout.txt", "w")
outfile.write(str(answer))
outfile.close()
```

Where to learn more Python, for free!



www.udacity.com

Try ***Intro to Computer Science*** with Dave Evans. You will create your own webcrawler and implement Google's *PageRank* algorithm!

Note: This course is using Python 2, so you will use `print` instead of `print()`, `raw_input()` instead of `input()`, and division looks confusing at first since $5/2 = 2$ in Python 2 whereas $5/2 = 2.5$ in Python 3.



Udacity is an incredible collection of free courses for programmers at all levels – go online and explore! (Click on *Catalog* and then search the *Free Courses*.)



www.edx.org

Try ***Introduction to Computer Science and Programming Using Python*** with renowned MIT professor Eric Grimson. The CS theory and coding in Python 3 is expertly explained to you on some highly entertaining videos!

Note: The course looks like it costs \$75, but you can actually choose to study for free! (without getting a certificate ...)

Next start date: **22 Jan 2019**.



There are many more tremendous courses available here – go online and have a look!!



www.coursera.org

Try ***Learn to Program: The Fundamentals*** with Jennifer Campbell and Paul Gries from Toronto University. This course introduces the fundamental building blocks of programming and teaches you how to write fun and useful programs in Python 3.

Note: The course looks like it costs \$67, but you can actually choose to study for free! (without getting a certificate ...)

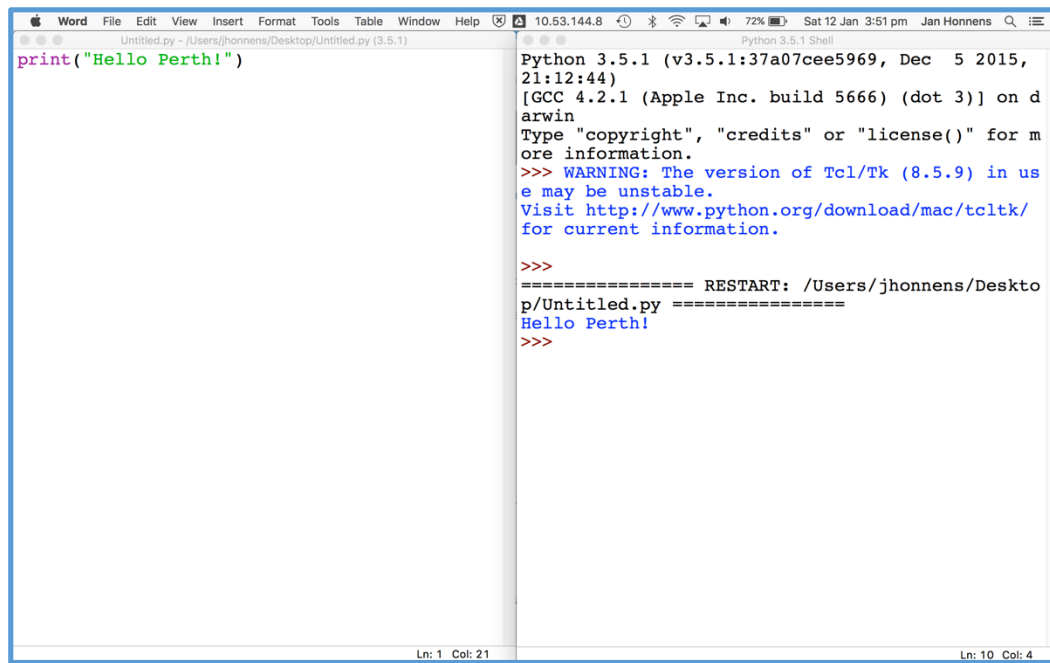


There are an abundance of fantastic courses available here – go online and have a look!

Installing Python on your computer, \$0 😊

A simple installation including all the standard modules like *turtle*, *random* and *math*:

1. Go to www.python.org, download and install the latest version of Python 3.
2. Open a new file in IDLE and organise your desktop as shown below.
3. Code!



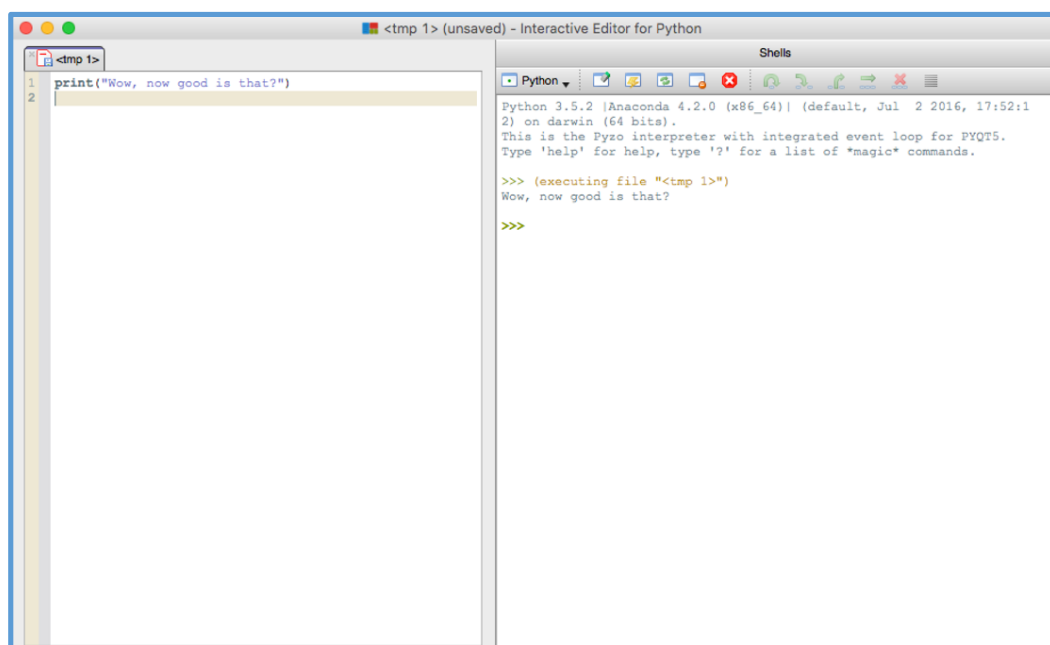
The screenshot shows a Mac OS window titled "Python 3.5.1 Shell". The left pane shows a file named "Untitled.py" with the code `print("Hello Perth!")`. The right pane shows the shell output, which includes the Python version (3.5.1), GCC version (4.2.1), and a warning about the Tcl/Tk version (8.5.9). The output also shows a restart command and the successful execution of the script, displaying "Hello Perth!".

```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec 5 2015, 21:12:44)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> WARNING: The version of Tcl/Tk (8.5.9) in use may be unstable.
Visit http://www.python.org/download/mac/tcltk/ for current information.

>>>
===== RESTART: /Users/jhonnens/Desktop/Untitled.py =====
Hello Perth!
>>>
```

An advanced installation including the libraries *numpy*, *scipy* and *matplotlib*:

1. Go to www.python.org, download and install the latest version of Python 3.
2. Go to www.continuum.io, download and install the latest version of Anaconda.
3. Go to www.pyzo.org, download and install the latest version of Pyzo.
4. Open a new file in Pyzo.
5. Code!



The screenshot shows a Mac OS window titled "<tmp 1> (unsaved) - Interactive Editor for Python". The left pane shows a file named "<tmp 1>" with the code `print("Wow, now good is that?")`. The right pane shows the shell output, which includes the Python version (3.5.2), Anaconda version (4.2.0), and the Pyzo interpreter information. The output also shows the successful execution of the script, displaying "Wow, now good is that?".

```
Python 3.5.2 [Anaconda 4.2.0 (x86_64)] (default, Jul 2 2016, 17:52:12) on darwin (64 bits).
This is the Pyzo interpreter with integrated event loop for PYQT5.
Type 'help' for help, type '?' for a list of *magic* commands.

>>> (executing file "<tmp 1>")
Wow, now good is that?
>>>
```