

# Python for SOI

Swiss Olympiad in Informatics, Workshops Fall 2017

## 1 IDE

PyCharm Community Edition (<https://www.jetbrains.com/pycharm/>)

## 2 I/O

```
string = input() # text
number = int(input()) # integer number
print(string) # appends newline
print(number, string) # separated by space
# {} will be replaced by arguments in order of appearance
print(f"Case_{testcasenumber}:_{solution}")
```

## 3 Files

```
import sys
sys.stdin = open("sample01.in") # open file to read
sys.stdout = open("sample01.out", "w") # open file to write
a = int(input()) # reading single numbers just like before
a, b, c = map(int, input().split()) # read 3 numbers from 1 line
```

## 4 Loops

```
for i in range(5): # for loop
    print(i) # prints 0, 1, 2, 3, 4
for i in range(2, 14, 4): # start, stop, step
    print(i) # prints 2, 6, 10

i = 5
while i > 0: # while loop
    i -= 1
```

## 5 List

```
numbers = [] # create new list
for i in range(5):
    numbers.append(input()) # append new element
for i in numbers: # iterate over list
    print(i)
print(numbers[2]) # access elements
print(len(numbers)) # list length
```

## 6 Terminal (in case you ever need it)

### Some useful commands

- `ls` 'list' (Windows: `dir`): show content of current directory (`ls -l` shows more details, `ls -a` includes hidden files)
- `cd path` 'change directory': change current directory to 'path'
- `cat file1 file2 ...` 'concatenate' (Windows: `type`): show contents of file1, file2, ...
- `pwd` 'print working directory': show current directory
- `diff file1 file2` 'difference' (Windows: `fc` 'file compare'): compare file1 and file2

Redirect output to a file: `python3 task.py > outfile.txt`

Send file to program: `python3 task.py < infile.txt`

Do both: `python3 task.py < infile.txt > outfile.txt`

Pipe output of program to other program: `python3 task.py | grep -e 'hello'`

### Paths (might vary on Windows)

Absolute paths: e.g. `/home/stofl/soi`

Relative paths start with `.` or `..`

- `.` means current directory
- `..` means parent directory

Relative to home: e.g. `~/soi`

### Keyboard shortcuts (might vary on Windows)

- TAB: auto-completion instead of typing everything
- Arrow-up/down: navigate through commands history
- Ctrl-R: search commands history
- Ctrl-Shift-C / Ctrl-Shift-V: copy/paste
- Ctrl-C: kill current program

## 7 Running Python on command line

Check with `python --version` that you are running Python 3.x. If `python` is Python 2.x, use `python3` to run your programs.

`python addition.py`

`python addition.py < infile.txt > outfile.txt`