



Persuasive Technologies: 01—Introduction to Python

Klen Čopič Pucihar

Program ponedeljek

1. Prpričljive tehnologije (1 ura)
2. Uvod v Python (1 ura)
3. Uvod v Kivy (2 uri)
4. HTTP Com Service (30 min)
5. Chat aplikacija(1 ura 30 min)

Alternativa:

<http://inclem.net/pages/kivy-crash-course/>

Enteris: 3, 10, 12, 14



What is Python?

- General-purpose scripting programming language

Beautiful is better than ugly.
Explicit is better than implicit. **Simple**
is better than complex. **Complex** is better
than complicated. **Flat** is better than
nested. **Sparse** is better than dense.
Readability counts. *Special cases* aren't
special enough to
break the rules.

Although **practicality** beats purity. *Errors* should never
pass silently. Unless **explicitly** silenced. In the face of
ambiguity, **refuse** the temptation to guess. There should be **one**
— and preferably only one — obvious way to do it. Although that
way may not be obvious at first *unless you're Dutch*. **Now** is
better than never. Although never is **often** better than *right*
now. If the implementation is *hard* to explain, it's a **bad**
idea. If the implementation
is *easy* to explain, it
may be a **good** idea.

Namespaces are
one *honking great*
idea — let's do
more of those!

Beautiful is better than ugly.
Explicit is better than implicit. **Simple**
is better than complex. **Complex** is better
than complicated. **Flat** is better than
nested. **Sparse** is better than dense.
Readability counts. *Special cases* aren't
special enough to
break the rules.
Although **practicality** beats purity. *Errors* should never
pass silently. Unless **explicitly** silenced. In the face of
ambiguity, **refuse** the temptation to guess. There should be **one**
— and preferably only one — obvious way to do it. Although that
way may not be obvious at first *unless you're Dutch*. **Now** is
better than never. Although never is **often** better than *right*
now. If the implementation is *hard* to explain, it's a **bad**
idea. If the implementation
is *easy* to explain, it
may be a **good** idea.

Namespaces are
one *honking great*
idea — let's do
more of those!

Why Python

- Ranked among the top eight programming languages
- Versatile
- Simplistic
- Compile not required

Where can you run Python scripts?

- Any system with Python interpreter



Python Interactive mode

```
$ python
>>> the_world_is_flat = True
>>> if the_world_is_flat:
...     print("Be careful not to fall off!")
...
Be careful not to fall off!
```

Python File

```
#!/usr/bin/env python  
# Hello World program in Python  
print ("Hello World!")
```

```
$ python 11_primer_1.py
```

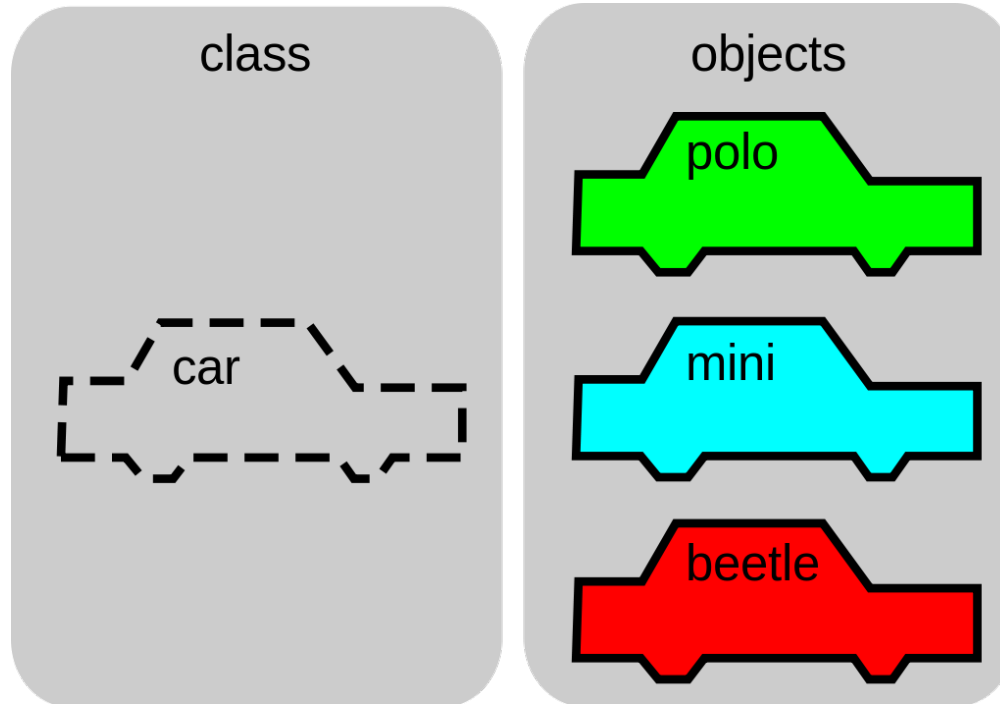
or

```
$ ./11_primer_1.py
```


Basics Python

- Aritmetics: nubers and strings
- Aritmetics: nubers and strings
- Assigmnet and while
- Flow Control: if and for
- Functions

Object Oriented Python



- What are attributes?
- What are methods?

Python glossary 1

- **Argument:** A value passed to a function (or method) when calling the function.
- **Function:** A series of statements which returns some value to a caller.
- **Expression:** A piece of syntax which can be evaluated to some value.

Python glossary 2

- Class: class is a blueprint. It isn't something in itself, it simply describes how to make something.
- Object: Any data with state (attributes or value) and defined behavior (methods).
- Attribute: A value associated with an object which is referenced by name using dotted expressions. For example, if an object o has an attribute a it would be referenced as o.a.
- Method: A function which is defined inside a class body.
- Module: An object that serves as an organizational unit of Python code. Modules have a namespace containing arbitrary Python objects. Modules are loaded into Python by the process of importing.

Coding style

- Use 4-space indentation, and no tabs.
- Wrap lines so that they don't exceed 79 characters.
- Use blank lines to separate functions and classes, and larger blocks of code inside functions.
- When possible, put comments on a line of their own.
- Use docstrings.
- Use spaces around operators and after commas, but not directly inside bracketing constructs: `a = f(1, 2) + g(3, 4)`.
- Name your classes and functions consistently; the convention is to use CamelCase for classes and `lower_case_with_underscores` for functions.
- Don't use fancy encodings if your code is meant to be used in international environments. Python's default, UTF-8.

Naloga I

- Naloga 1: Napisi python skripto, ki od uporabnika zahteva dve stevil. Stevili seštej in izpisi rezultat.
- Naloga 2: Napisi python skripto, ki polk dveh stevil omogoča uporabniku izbrati tudi racunsko operaijo.