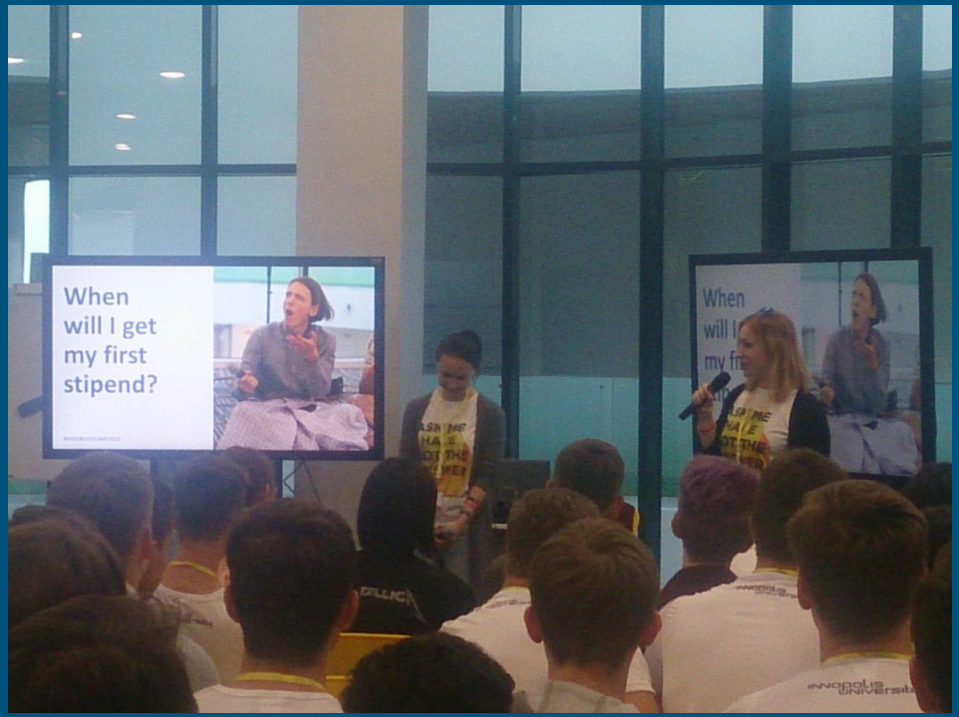
Programming fundamentals I

Lab by Artem Bakhanov

I am Artem (Artyom)



Innopolis changes people

what about you?

python

why python?

Popular language! (3rd position in TIOBE index - https://www.tiobe.com/tiobe-index/)

Big standard library

The huge set of third-party libraries

A lot of language-level data structures

Easy to start, easy to learn

python is...

The high-level language

OOP language

Functional language

Modular language

Dynamically typed language

Interpreted language

cons

Python is slow

Python is not a very good language for mobile development.

Python is not a good choice for memory intensive tasks.

It's near impossible to build a high-graphic 3D game using Python.

Has limitations with database access.

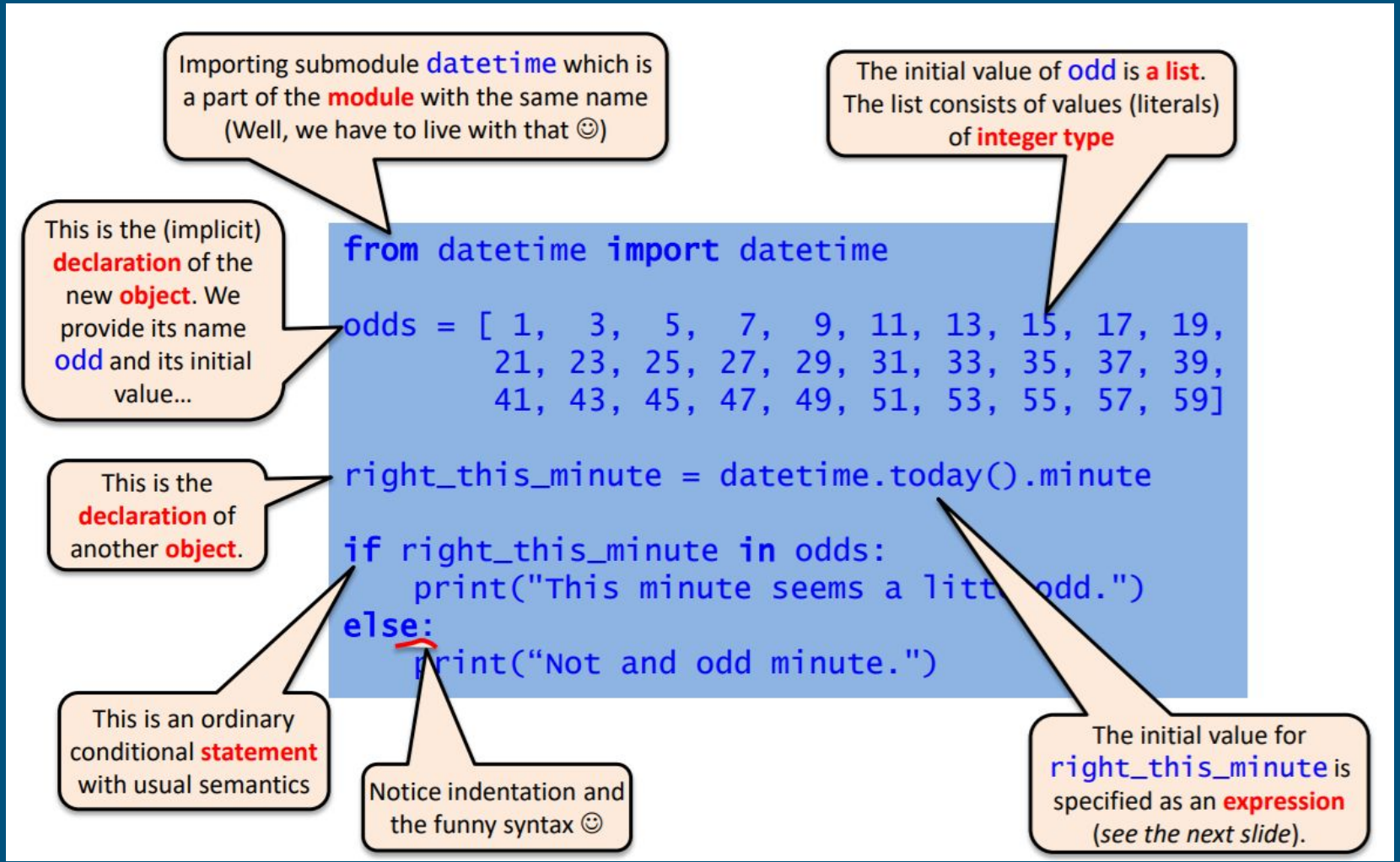
Python is not good for multi-processor/multi-core work.

main applications of python

Web Development

Data Science — including machine learning, data analysis, and data visualization

Scripting


Importing submodule datetime which is a part of the module with the same name (Well, we have to live with that ☺)
The initial value of odd is a list. The list consists of values (literals) of integer type
This is the (implicit) declaration of the new object. We provide its name odd and its initial value...
from datetime import datetime
odds = [1, 3, 5, 7, 9, 11, 13, 15, 17, 19,
21, 23, 25, 27, 29, 31, 33, 35, 37, 39,
41, 43, 45, 47, 49, 51, 53, 55, 57, 59]
right_this_minute = datetime.today().minute
if right_this_minute in odds:
print("This minute seems a litt odd.")
else:
print("Not and odd minute.")
This is the declaration of another object.
This is an ordinary conditional statement with usual semantics
Notice indentation and the funny syntax ☺
The initial value for right_this_minute is specified as an expression (see the next slide).

Tasks

- Write a program that prints ‘Hello World’ to the screen.
- Write a program that asks the user for a number n and prints the sum of the numbers 1 to n
- Write function that reverses a list, preferably in place.
- Write a function that tests whether a string is a palindrome.
- Write a function that finds all prime numbers between 1 and N

Create a program that determines the complexity of a given password based on these rules:

- A very weak password contains only numbers and is fewer than eight characters.
- A weak password contains only letters and is fewer than eight characters.
- A strong password contains letters and at least one number and is at least eight characters.
- A very strong password contains letters, numbers, and special characters and is at least eight characters.

Example Output

```
The password '12345' is a very weak password.
The password 'abcdef' is a weak password.
The password 'abc123xyz' is a strong password.
The password '1337h@xor!' is a very strong password.
```

Constraints

- Create a `passwordValidator` function that takes in the password as its argument and returns a value you can evaluate to determine the password strength. Do not have the function return a string—you may need to support multiple languages in the future.

A password that does not meet all complexity requirements is “average”

Task: Create a simple console expression calculator

- Binary operations
  - Addition
  - Subtraction
  - Division
  - Multiplication
- Unary negation operator
- Parenthesis support
- Sample expression -((5+2.2)*4+2.5)/2

Recommended step-by-step approach:

1. Tokenize the expression.
2. Define the grammar.
3. Prepare the syntax tree.
4. Traverse the syntax tree (calculate the expression).
5. Create REPL.