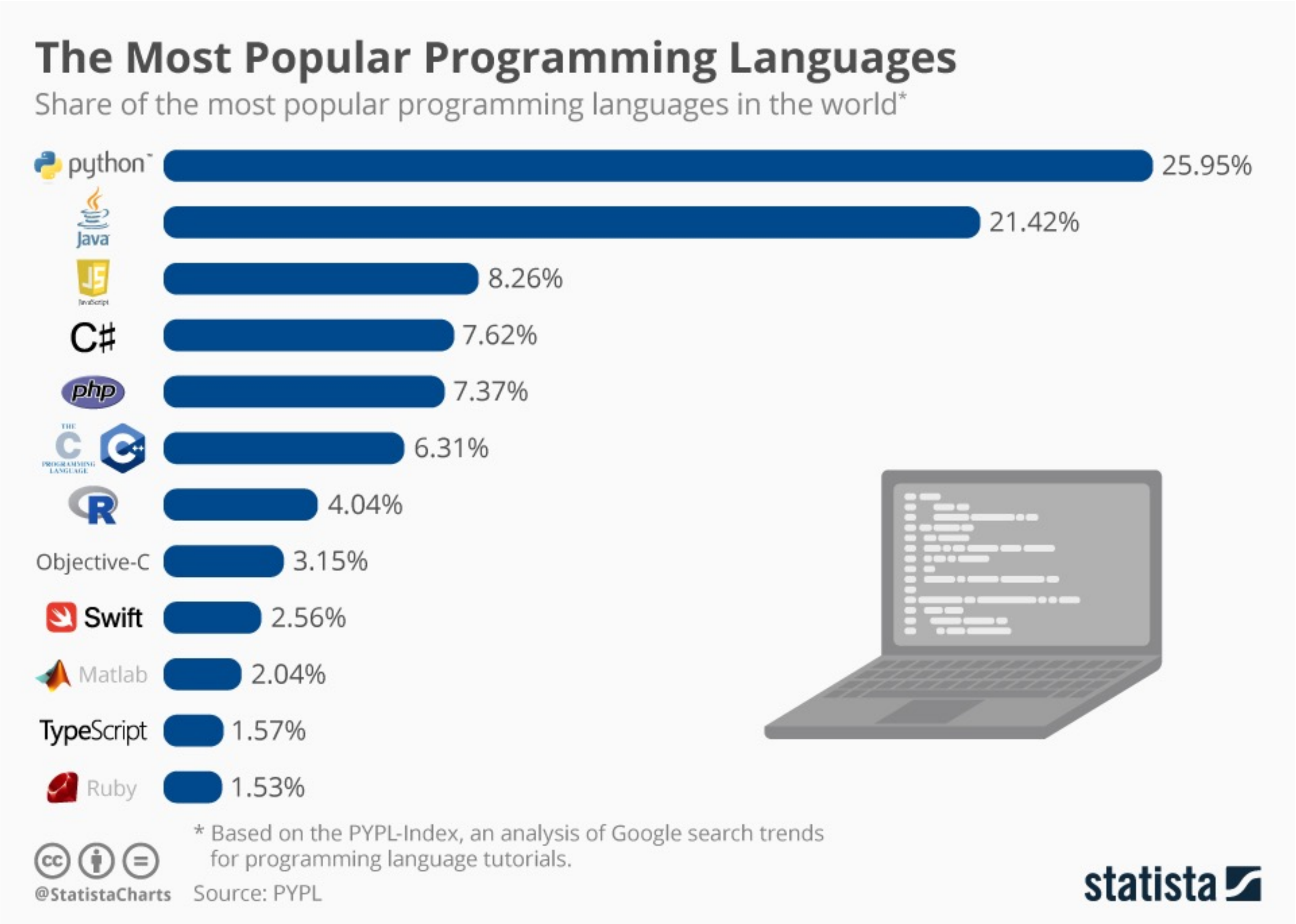
Seminar 01

Python programming 1

(installation, simple calculation, variables)

Programming languages



The Most Popular Programming Languages

Share of the most popular programming languages in the world*

* Based on the PYPL-Index, an analysis of Google search trends for programming language tutorials.

@StatistaCharts Source: PYPL

Some of the Main features of Python

✓high-level programing language

✓general-purpose programming language

✓open-source (free of charge)

✓availability of a lot of high-quality packages

✓good readability

× interpreted language which inherently makes it slow

× forces you to indent your code

Installing python on a computer

- There are two common approaches to install python:

Installing python and required libraries and software separately.

Python interpreter from python.org

- Python interpreter
- Standard library

Install using available bundles

Python distribution (Anaconda as example)

- Python interpreter
- Standard library
- Addition packages
- Useful applications for coding
  - Ipython
  - Integrated Development Environment (e.g. Spyder)
  - Jupyter notebook
- Environment management system (Conda)

…and more

https://www.anaconda.com/download

Writing and Running Python Code

There are different ways to run Python code:

- Use a Python shell (standard Python shell/IPython shell)
  - type in command prompt: `python` `or` `ipython`
- Run a Python script (`xxxx.py`) from command prompt
  - type in command prompt: `python example.py`
- Use a Python IDE
  - e.g. Spyder
- Use Jupyter notebook

… and more

Language Syntax: Variables

A Python variable is **a reserved memory location to store values**. In other words, a variable in a python program gives data to the computer for processing.

Names of variables:

✓ `A = 1`

✓ `B = 4`

✓ `number_one = 14`

✓ `Kappa1 = 11`

*First character of variable names cannot be a number

× `1kappa = 11`

Language Syntax: Statements and Assignments

```
variable(s) = expression(s)
```

- Left hand side can **not** involve operation

  ×
  ```
  x + y = 1
  ```

- Serialized assignment

  ✓
  ```
  n1 = n2 = n3 = 1
  ```

- Stacked assignment

  ✓
  ```
  a, b = 3, 4
  print(a)
  > 3
  print(b)
  > 4
  ```

- Line break

  ✓
  ```
  a = 1*1 + 2*2 + 3*3 + 4*4 + 5*5 + 6*6 + 7*7 + 8*8 + 9*9 + 10*10 + 11*11 + 12*12
  ```

  ✓
  ```
  a = ( 1*1 + 2*2 + 3*3 + 4*4 + 5*5 + 6*6 + 7*7 +
        8*8 + 9*9 + 10*10 + 11*11 + 12*12 )
  ```

- Delete statement

  ```
  del a
  ```

Language Syntax: Variable Types

Variable Types	Integers	Floating Point Numbers	Boolean	Complex numbers	Strings
Information stored	An integer	A real number	can only either be True or False	A complex number	A “string” of letters
Example	a = 1 type(a) > Int	a = 1.0 type(a) >float	a = True type(a) > bool	a = 1.0 + 1.0j type(a) > complex	a = "Hello" type(a) > str

Language Syntax: Integer, Float and Complex

- Announce an integer:

```
a = 5
type(a)
> int
```

- Announce an integer as `float` type

```
a = 5.
type(a)
> float
```

- Scientific representation is allowed for `float`

```
a = 1e-2
a = 5.12E15
```

- Calling real or imaginary part of complex number

```
x = 1.0 + 2.0j
x.real
> 1.0
x.imag
> 2.0
```

Language Syntax: Strings

- Strings can be quoted either with ‘ or “

```
s = 'a string'
s = "a string“
```

- Long strings with line breaks can be quoted by triple quotation

```
s = '''a string with
       line breaks'''
s = """a string with
       line breaks"""
```

Language Syntax: Type conversions

```
int(x), float(x), complex(x), bool(x) and str(x)
```

E.g.

```
a = 1
b = float(1)
type(b)
> <type 'float'>
```

Language Syntax: Functions

- In **Python**, a **function** is a group of related statements that performs a specific task.
- Functions can be called by appending round brackets to them
- Anything you write in the brackets is the argument of the function and it will perform some operation with this data.

E.g.

```
a = 'hello'
print(a)
> Hello
```

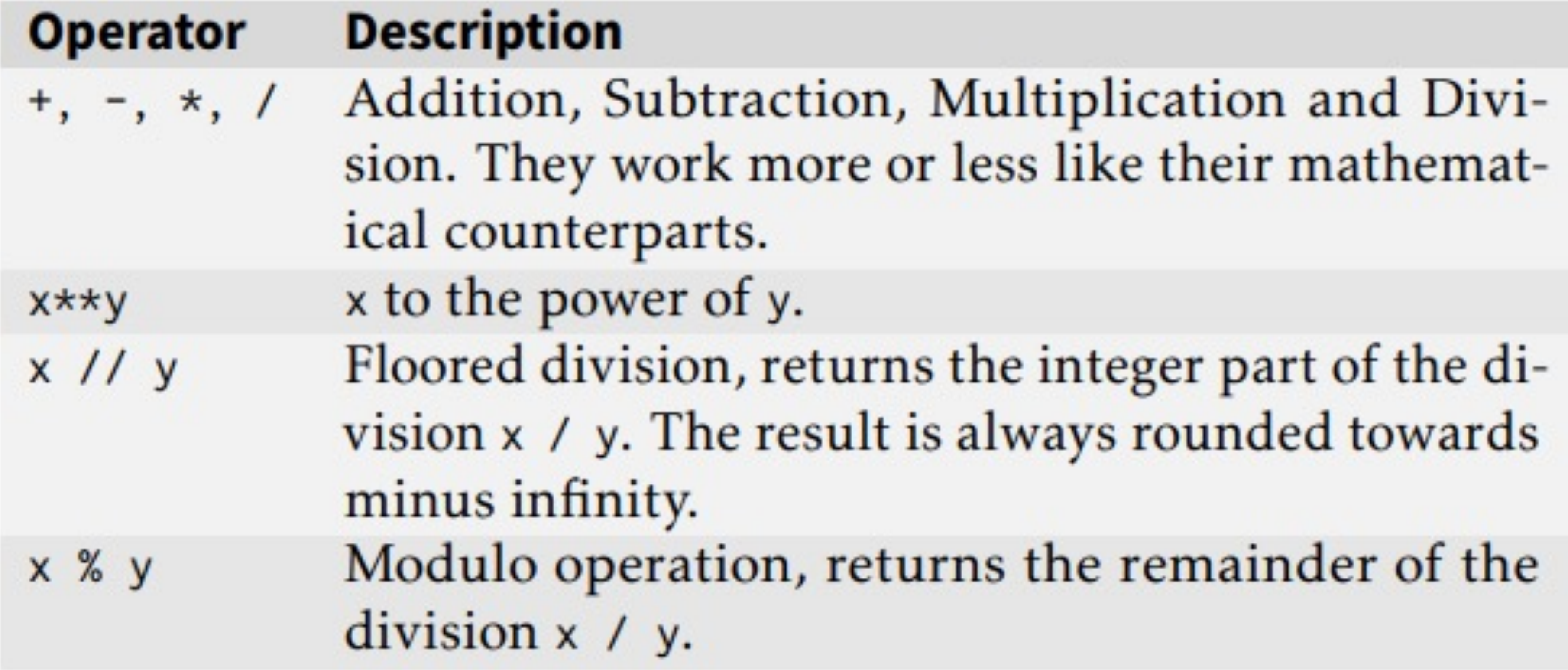
Function	`print()`
Argument	a
Operation performed	Printing the argument in command prompt/shell

- Functions can also have a return value.

E.g. `b = float(1)` b is the return value of function `float()`.

- Depending on the function they can accept several arguments separated by a comma.

Operators on Numerical Types



Operator	Description
`+, -, *, /`	Addition, Subtraction, Multiplication and Division. They work more or less like their mathematical counterparts.
`x**y`	`x` to the power of `y`.
`x // y`	Floored division, returns the integer part of the division `x / y`. The result is always rounded towards minus infinity.
`x % y`	Modulo operation, returns the remainder of the division `x / y`.

Augmented Assignments:

- ✓ x = x + 1
- ✓ x += 1

Also `-=`, `*=`, `**=`, etc.

Operators on Strings

Operator	Function	Example

+	concatenation	`'python' + ' ' + 'course'`
		`> 'python course'`

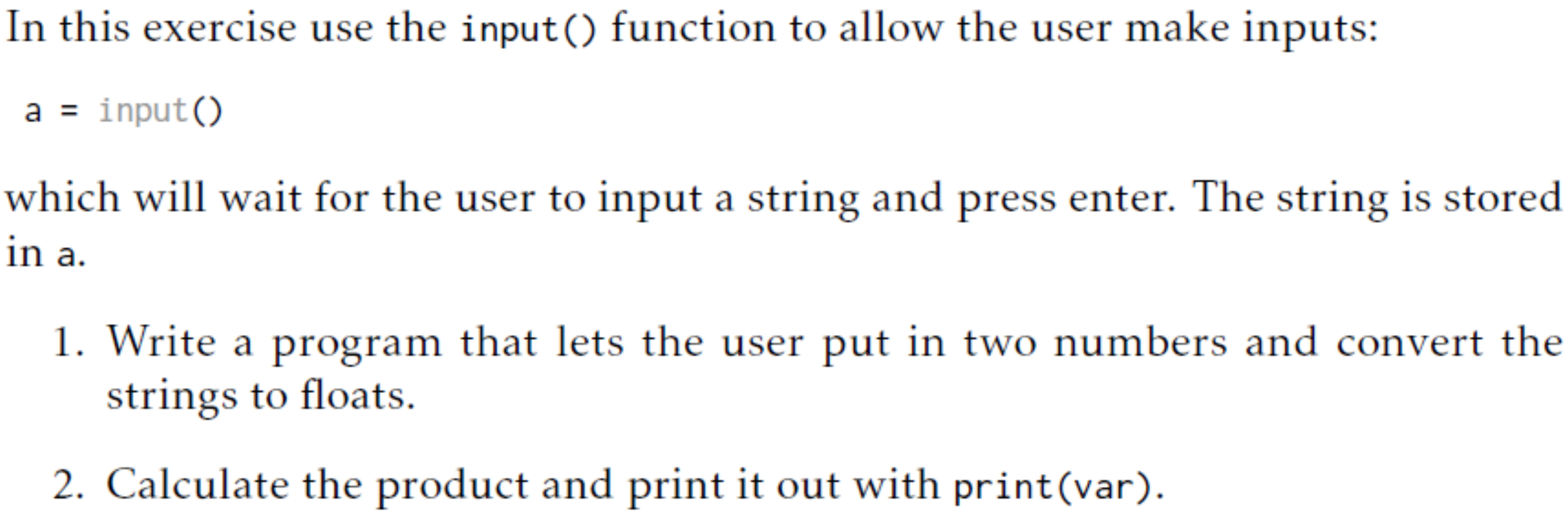
-
/
**
//
%
	not applicable	`'python' - 'course'`
		`> TypeError`
		`> TypeError: unsupported operand type(s) for -: 'str' and 'str'`

*	replicate	`'py' * 5`
		`> 'pypypypypy'`

Exercise 1

- 1. Create a new Python script.
- 2. Define several variables and assign them numbers as content.
- 3. Store the sum of these variables in another variable.
- 4. Now print the sum and two variables with the function print().

Exercise 2



In this exercise use the input() function to allow the user make inputs:

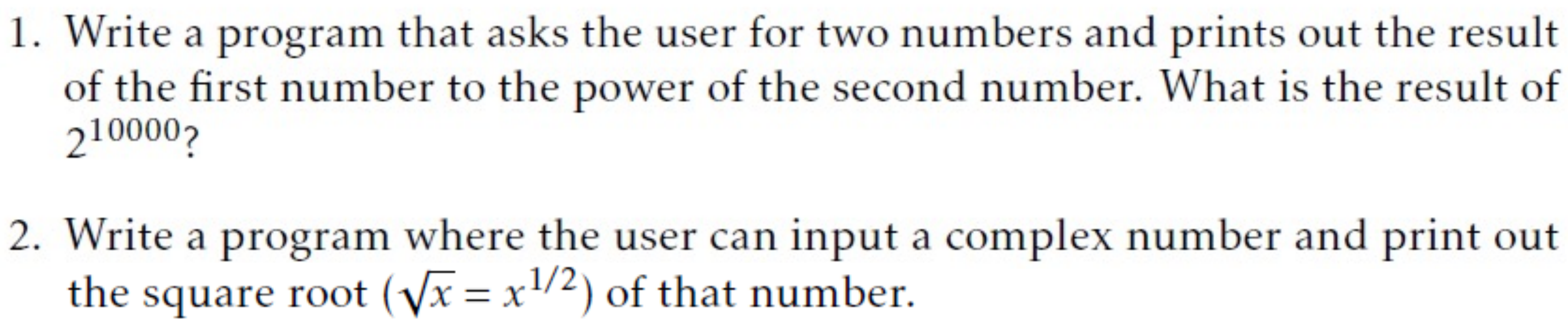
```
a = input()
```

which will wait for the user to input a string and press enter. The string is stored in a.

1. Write a program that lets the user put in two numbers and convert the strings to floats.

2. Calculate the product and print it out with print(var).

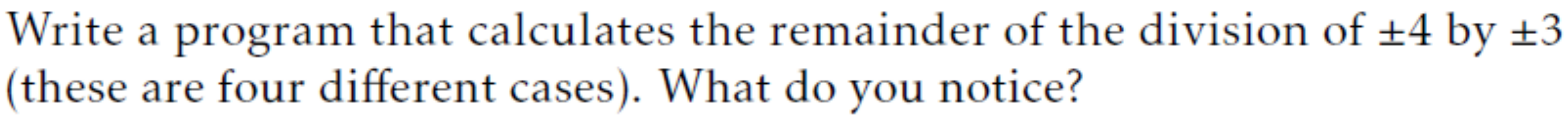
Exercise 3



1. Write a program that asks the user for two numbers and prints out the result of the first number to the power of the second number. What is the result of 2^{10000}?

2. Write a program where the user can input a complex number and print out the square root ($\sqrt{x} = x^{1/2}$) of that number.

Exercise 4



Write a program that calculates the remainder of the division of ±4 by ±3 (these are four different cases). What do you notice?