

# Python Start

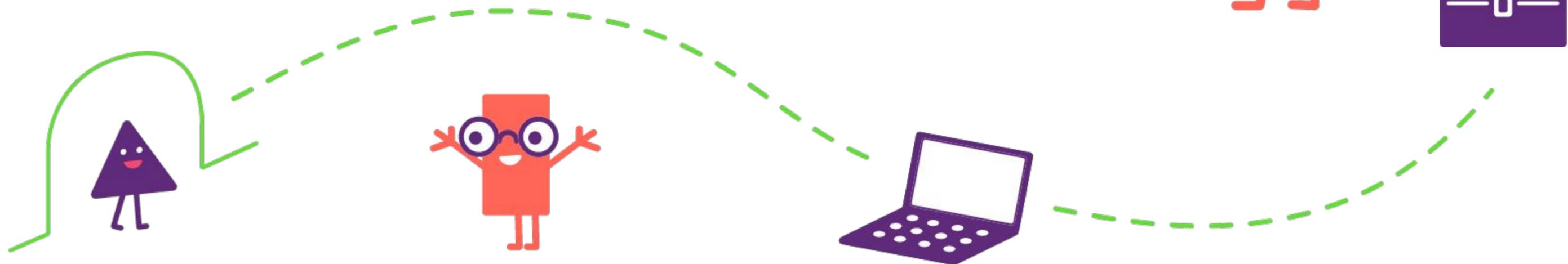
A course for kids aged 11-13

Fun programming  
in one of the world's  
most popular languages



# A solid start in programming

Kids learn to code in Python – one of the most popular programming languages in the world. They get to apply their knowledge in practice, creating programs for study, entertainment and everyday life



# Everyone will enjoy it!



## A deep dive into the process

The course uses a storyline about working in a real IT company, which helps us retain the attention of students who find concentrating difficult



## We nurture mathematical thinking

By studying the basic principles of programming, we deepen students's knowledge of math, even if they've never done well in the subject before



## We find their motivation

We don't do tests, instead we apply what we've learned in practice straight away, by creating projects and bringing our own little dreams to life

# In their first year of studies, teens will learn to:



- ◆ Understand the basics of algorithms and object-oriented programming
- ◆ Develop interactive graphical games for PCs using the PyGame library
- ◆ Work with graphics and use the Turtle library
- ◆ Solve real tasks using Python, and apply an iterative approach
- ◆ Apply the principles of project work when creating projects

# In their second year of studies, **teens** will learn to:

- ◆ Write and read code in Python and work with data structures
- ◆ Develop sophisticated games and apps for PCs using the PyGame and PyQt libraries
- ◆ Design interfaces
- ◆ Automate work with graphical files
- ◆ Work in a team and create projects from ideas before publicly presenting them



# Why Python?

- It's one of the most popular and universal programming languages according to dozens of rating systems
- It's widely applied in various spheres of activity – from solving simple tasks to artificial intelligence and machine learning
- Knowledge of Python is in highly sought-after among job applicants to large IT companies, as well as other employers
- Programmers working in Python are some of the most highly paid specialists in the world



# Course storyline

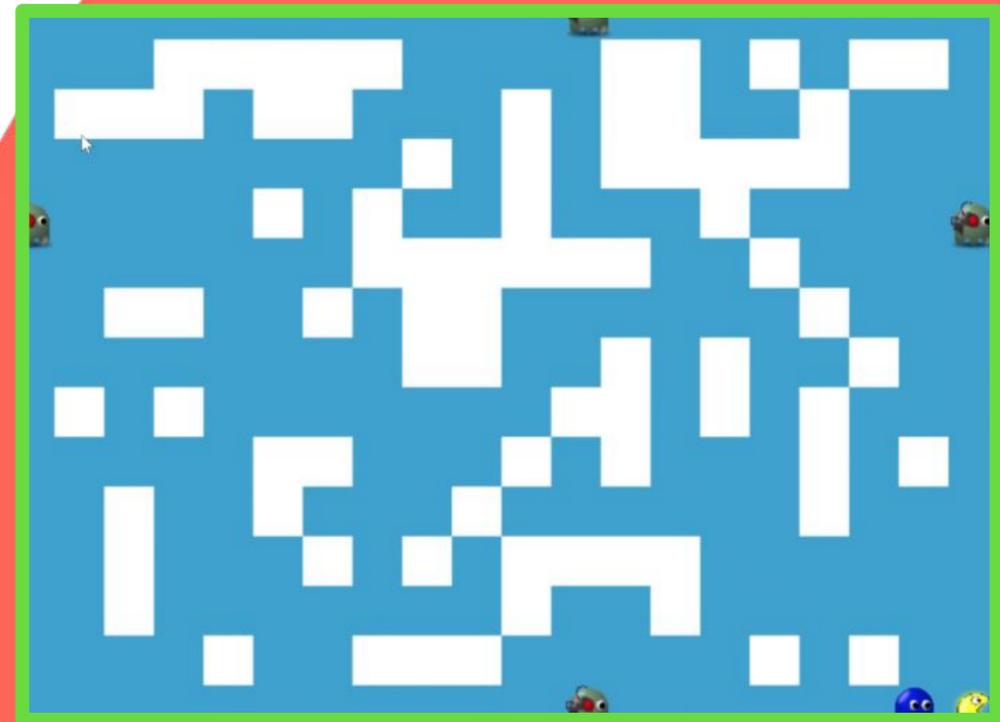
At the start of their studies, each student signs an improvised job contract making them a trainee at an IT company. They then move up the career ladder to the position of senior Python developer.

The storyline introduces teens to such concepts as **employers, projects, brainstorming, deadlines, time management, checklists** and **mind maps**, and ties together all the projects and tasks on the platform.



# A project-based approach

- ◆ Our kids create mini-projects right from their first lessons, applying the knowledge they've gained in practice
- ◆ Kids share their projects with their classmates directly in the platform, and learn to give and receive feedback
- ◆ At the end of each module, they present a full individual or group project



# What are our classes like?

- **Online** or **at the Algorithmics** school in your city
- In groups of **up to 10** online and up to **12** offline
- Classes last for **90 minutes** with a break in the middle
- **Once a week**,  
2 academic years

The teacher explains the material in an interesting way and **gets the kids interested in the new topic**

Your kid won't ever fall behind in the program: **any classes they miss can be taken on the platform, 24/7**

You won't need to check any homework: at Algorithmics, **there are no obligatory homework tasks**

You'll be given **access to the platform** and will be able to follow your kid's progress

# Course structure – 1<sup>st</sup> Year 3 Levels

## Beginner

### Module 1. Language basics

- Introduction to Python. Input-output functions.
- Variable. Numerical data types
- Strings
- Nested constructs

### Module 2. Control structures

- The logical type of data. Conditional statements
- Nested conditional statements and conditional statements with severe branches
- The “while” loop. Loops with a counter.
- Nested algorithmic constructs (loops in loops, conditional statements in loops, etc.)

### Module 3. Functions and modules

- Creating functions. Local and global scopes
- Using one function inside another
- Modules. Using nested modules from the standard library
- Creating modules. Connecting multiple modules

## Intermediate

### Module 4. The Turtle module. Math for developers

- The Turtle graphical module. The math behind the module’s movement
- Conditional statements. Drawing simple geometric shapes
- Loops. Drawing polygons
- Project exercise: graphic project

### Module 5. Object-oriented programming

- Objects and their fields and methods
- Handling mouse and keyboard events
- Classes. The class constructor
- Inheritance (from a ready-made or one’s own class)

## Advance

### Module 6. Basics of game development in PyGame\*

- Basics of game development. Sprites and the game loop
- Lists and methods of working with them. Iterating through lists in the “for” loop
- Handling in-game events
- Game physics
- Full Recap

\*Only in the full version of the course



# Course structure – 2<sup>nd</sup> Year 3 Levels

## Beginner

### Module 1. Data structures

- Recap
- Data structures: lists and dictionaries
- Nested data structures
- Handling exceptions

### Module 2. Developing windowed applications

- Main widgets of a windowed application
- Designing an app interface
- Creating the professional computer app “Memory Card”

### Module 3. Working with files

- Recording and reading data from text files
- Recording and reading data from JSON files
- Creating the windowed application “Smart Cards” with record search by tags

## Intermediate

### Module 4. Automatic image processing

- The PIL library (Python Imaging Library)
- Correcting the size, placement, brightness, color and contrast of images
- Mass processing of images

### Module 5. Advanced game development in PyGame\*

- Different types of sprites and their properties
- Using the physical properties of sprites
- Game event handling
- Designing a game: background and music

## Advance

### Module 6. The developer’s portfolio\*

- Making creative programs and apps. Project presentation

\*Only in the full version of the course



# Why do people choose Algorithmics?

- ◆ The curriculums for all our courses are developed by a team of professional **educators, pedagogues and psychologists**
- ◆ Algorithmics' **teachers** talk to the kids in understandable language, love their subject and know how to captivate children
- ◆ Our **IT learning platform** is 3 in 1: it's a smart task book, an environment for creating projects, and a community of shared interests



# Algorithmics

International School of Programming  
for children aged 6 to 17

😊 860 000 graduates

🚩 80 countries

🏠 450 partners



# Courses for kids aged 6-17

Kids can start studying at Algorithmics at any age. At the end of the course, students can move straight on to the next one to continue studying in the new academic year

Course name	Age:	5 – 6	7 – 9	10 – 11	12 – 13	14 – 15	16 – 17
Python Pro (2 years)							
<b>Python Start (2 years)</b>							
Game Development on Unity							
Game Design							
Visual Programming							
The Coding Knight							

Algorithmics

**Book a place in one  
of our groups**

<http://algorithmicschool.nl/>

