



Курс программирования Python Pro

Подготовка к профессиям будущего
через освоение лучших практик
программирования на Python

14-17
лет



алгоритмика

Международная школа
программирования
для детей

Почему выбирают наш курс

Углубленно изучаем программирование

Обучаем ребят профессиональным инструментам и практикам программирования на Python.

Развиваем проектное мышление

Ребята составляют план, распределяют задачи, публично защищают проект. Учатся проверять себя, исправлять ошибки, в том числе в чужом коде.

Готовим к профессиям будущего

Навыки программирования, системное мышление, создание проектов и умение работать в команде пригодятся для успешной карьеры в любой сфере.





По окончании курса ваш ребенок сможет:

- профессионально программировать на языке Python
- анализировать и визуализировать данные
- писать программы и игры разной сложности
- понимать в каких профессиях можно успешно применять навыки программирования на Python
- решать задачи, которые выдают для прохождения собеседования в крупнейших компаниях

Будет интересно всем!

- Творческим натурам — программировать цифровые интерактивные шедевры;
- Будущим инженерам — проектировать сложные технические системы и процесс их монтажа;
- Юным ученым — моделировать эксперименты, доказывая жизнеспособность новых теорий;
- Будущим управленцам — анализировать данные и строить гипотезы для решения различных бизнес-задач;
- Перспективным программистам — начать свой карьерный путь в крупной IT-компании, показав в качестве начального портфолио проект, созданный в «Алгоритмике».



Чему научимся за год

Программирование:

- созданию сложных программ под различные задачи
- работе с библиотеками для Python разработчиков
- созданию динамических объектов
- использованию инструментов визуализации данных

Метапредметные навыки:

- применению междисциплинарного подхода для достижения результата
- решению задач и кейсов, которые дают при устройстве на работу крупные IT-компании

Углубление знаний школьной программы:

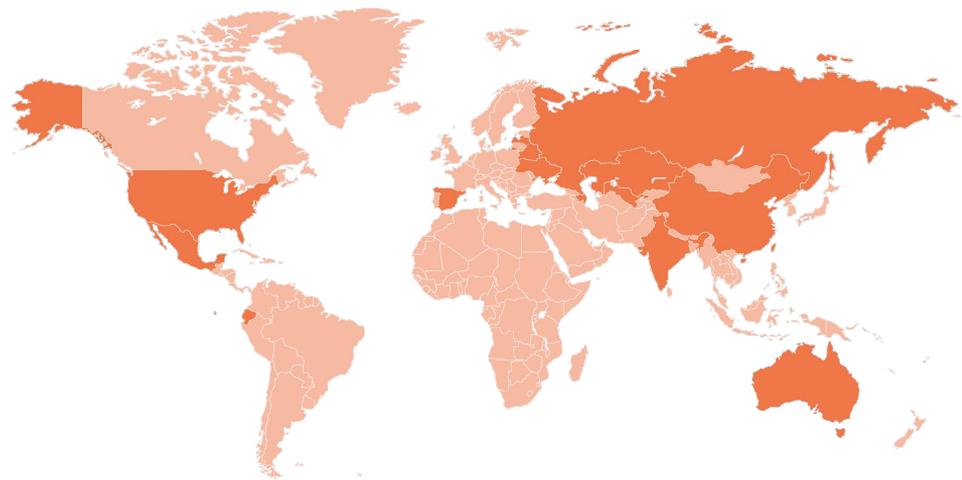
- новым словам и выражениям на английском языке
- новым знаниям по математике, физике и другим естественнонаучным дисциплинам
- техникам решения задач для успешной сдачи экзаменов

Как проходят уроки

- **групповые занятия 1 раз в неделю по выходным**
 - вспоминаем, что было на прошлом уроке
 - изучаем новую тему
- **продолжительность занятия 90 минут**
 - программируем, закрепляем материал
 - создаем программу, игру, приложение, дополняя проект новыми инструментами; обсуждаем в группе итоги занятия
- **индивидуальный подход к каждому ребенку**
 - создаем проект: формулируем идею, находим решение, программируем, усложняем
 - обмениваемся результатами и идеями, учимся читать чужой код и анализировать его. Даем обратную связь.

Алгоритмика — международная школа программирования для детей от 5 до 17 лет

- Собственная интерактивная образовательная платформа, позволяющая каждому ребенку осваивать знания в индивидуальном темпе
- Команда профессиональных методистов, педагогов и психологов, работающая над созданием лучших курсов программирования для детей



150

тыс. детей

200

городов

15

стран

Ученики «Алгоритмики»

- создают портфолио собственных проектов
- находят единомышленников и друзей, обмениваются идеями, помогают друг другу
- участвуют и побеждают в городских и региональных хакатонах и олимпиадах по программированию



Курсы международной школы программирования «Алгоритмика»



**Основы логики
и программирования**

5-7 лет



**Компьютерная
грамотность и цифровое
творчество**

7-12 лет



**Креативное
программирование**

8-12 лет



**Курс
программирования
Python Start**

11-13 лет



**Курс
программирования
Python Pro**

14-17 лет

Какие знания ребята получают за год

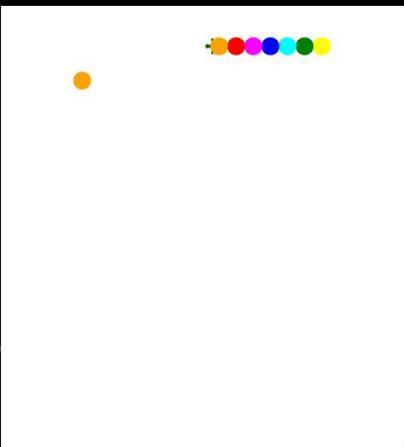
| | |
|--|---|
| <p>Модуль 1. Введение в Python</p> | <p>Модуль 2. Циклы и списки</p> |
| <ul style="list-style-type: none">• Понятия встроенных типов данных: строка, число• Программирование арифметических операций• Ввод данных: input() и print().• Понятие переменных и условного оператора: определение и грамотное использование, понимание применения условного оператора внутри условного оператора (вложенный условный оператор)• Написание программ: итерационный подход к написанию кода, следование стандарту оформления кода, нахождение ошибок в своей программе, поэтапная разработка с постоянным анализом результата. | <ul style="list-style-type: none">• Циклы: устройство циклов while и for, автоматизация рутинных процессов с помощью циклов, понимание применения условного оператора внутри цикла• Строки и списки: определение и навыки применения в программах, методы работы со строками• Структура данных: определение и навык выбора подходящей структуры данных исходя из условия• Алгоритмы сортировки, расчет сложности алгоритмов и графы• Понятие Big Data |

Какие знания ребята получают за год

| | |
|---|--|
| <p>Модуль 3. Функции и классы</p> | <p>Модуль 4. Проектный модуль</p> |
| <ul style="list-style-type: none">• Функции, объекты, свойства и методы: определения и особенности их применения, самостоятельное создание собственных функций и объектов• Модуль: определение, навык создания и подключения, назначение модуля random (задание случайных чисел), назначение модуля math (математические операции)• Класс: определение, понятие наследования классов, события, и умение это программировать• Программирование готовых объектов в своих программах и создание своих.• генерация баз данных | <ul style="list-style-type: none">• Разработка пользовательского интерфейса с помощью PyQt• Программирование чат-бота• Работа с API• Использование чужого кода: анализировать, изменение• Презентация собственных проектов |

Что мы будем создавать

```
1 from time import sleep
2 from random import randint
3
4 SCR_WIDTH = window_width()
5 SCR_HEIGHT = window_height()
6 TURTLE_SIZE = 10
7 delay = 1/10
8 step = 20
9
10 cellsX = round(SCR_WIDTH / (2*step)) - 1
11 cellsY = round(SCR_HEIGHT / (2*step)) - 1
12 prizeX = step * randint(-1 * cellsX, cellsX)
13 prizeY = step * randint(-1 * cellsY, cellsY)
14
15 STARTED = False
16 FINISHED = False
17
18 points = 0
19 nextDirX = 1
20 nextDirY = 0
21
22 colors = ['red', 'orange', 'yellow', 'green', 'cyan', 'blue']
23 curColor = 1
24 def nextColor():
25     global curColor
26     curColor = 0 if curColor > 5 else curColor + 1
27
28 def setNextDir(x, y):
```



```
1 # Processing sketch
2
3 theta = 0
4 length = 0
5
6 def setup():
7     size(300, 300)
8
9 def draw():
10    theta = theta + 0.01
11    background(0)
12    stroke(255)
13    strokeWeight(1)
14    # move to / movement_delta * dt
15    theta = radians(theta)
16    translate(cos(theta) * movement_delta, movement_delta * sin(theta))
17    translate(0, -length)
18    strokeLength(length)
19    rotate(theta)
20
21 def keyPressed():
22     key = key
23     if key == 'r':
24         reset()
25     if key == 'p':
26         print(x, y, theta)
27     if key == 'q':
28         quit()
29
30 def reset():
31     theta = 0
32     length = 0
33     strokeLength = 0
34     strokeWeight = 1
35     stroke(255)
36     rotate(theta)
37
38 def keyPressed():
39     key = key
40     if key == 'r':
41         reset()
42     if key == 'p':
43         print(x, y, theta)
44     if key == 'q':
45         quit()
46
47 # END
```



```
1 from processing import *
2
3 from random import randint
4 from math import pi, sin, cos
5
6 width = 300
7 height = 300
8
9 nodesTotal = 7 # число узлов
10 nodesDelay = 80 # время появления нового узла
11 curDelay = 0
12 elMinSize = 5 # размер итогового кружка
13 elMaxSize = 10
14
15 stripesMin = 3 # минимальное число полос от узла
16 stripesMax = 8 # максимальное число полос от узла
17 stripeLength = 50 # средняя длина полосы
18 stripeLengthD = 30 # разброс длин полос
19
20 exponent = 5 # кривизна линий
21
22 # user:
23 class Node():
24     def __init__(self, ):
25         self.x_center = randint(left, right)
26         self.y_center = randint(top, bottom)
27         self.step = randint(1, 3)
28         self.last_size = randint(elMinSize, elMaxSize)
29         self.lasting = randint(int(nodesDelay/4), nodesDelay)
30     # self.delay = nodesDelay
31
32 class Stripe():
33     def __init__(self, number, node, angle, distance, delta2=0):
34         self.node_number = number
35         self.start_x = node.x_center
36         self.start_y = node.y_center
37         self.cur_x = node.x_center
38         self.cur_y = node.y_center
39
40         self.step = node.step # - величина шаг
41         self.delta = (node.last_size - 1) * self.step / 100 # -
42         self.lasting = node.lasting # - время засты
43
44         a = angle
```

алгоритмика

**Бронируйте
место в группе!**