



Методические рекомендации по теме

«Модуль turtle: модель RGB, понятие объект, основные команды управления»

Цель:

- дать представление о программировании векторной графики в языке Python.

Задачи:

- знакомство с модулем turtle;
- программирование базовых графических объектов с помощью исполнителя на языке Python;
- анализ программного кода с целью определения, что выведет программа при конкретных исходных данных;
- исправление ошибок и дописывание программного кода;
- написание программного кода.

Планируемые результаты

Личностные: обучающиеся получают навыки активной коммуникации в группе, осознанной ориентировки в мире ИТ профессий, постановки собственных образовательных целей и задач, владение первичными навыками анализа и критичной оценки получаемой информации.

Предметные: обучающиеся получают представления об использовании модуля turtle в языке Python.

Метапредметные: обучающиеся получают возможность владения общепредметными понятиями «градусы», «исполнитель», «прямой угол», «векторная графика»; информационно-логическими умениями; умениями самостоятельно планировать пути достижения целей; владения умениями принятия решений и осуществления осознанного выбора; повышения уровня ИКТ – компетентности и расширение кругозора в области информатики и

программирования; знакомство с профессиональной деятельностью программиста в рамках ранней профориентации; развитие интеллектуальных способностей, а также логического и критического мышления.

Материалы к занятию

Приложение 1: Сценарный план видеоролика

Приложение 2: Домашнее задание и практика

Приложение 3: Краткие организационно-методические рекомендации по организации работы на занятии

Ход проведения урока

1. Организационный момент.

Мотивация на учебную деятельность.

Приветствие учащихся, сообщение темы и целей занятия.

2. Вводный блок.

Тема.

Преподаватель при необходимости останавливая трансляцию, комментируя дополнительно тему занятия.

Проблемная дискуссия по вопросам:

- Можем ли мы рисовать с помощью программирования?
- Как мы можем создавать графические объекты?
- Как вы понимаете фразу «рисовать с помощью программирования исполнителя»?
- Какие команды можно было бы использовать для программирования графических объектов?

Итоги дискуссии (обобщаются преподавателем и фиксируются ответы учеников на доске, чтобы вернуться к ним и оценить правильность предположений учеников на этапе рефлексии):

- Идеи по командам для создания графики
- Идеи по тому, как может использоваться Python для создания графических объектов.

**см. сцены 1 – 2 (здесь и далее приводится Таблица «Содержание видеоролика». Приложение 1).*

3. Блок повторения.

Блиц-опрос.

Преподаватель предлагает ученикам ответить на 5 вопросов по предыдущей теме; задания выполняются в сопровождении видеоролика с использованием таймера; ученики выполняют задания, голосуют, обсуждают результаты. Процедура голосования определяется инструкцией; учитель должен убедиться, что всем понятна процедура голосования. Преподаватель может поставить ролик на паузу и обсудить результаты голосования; объяснить правильный ответ руководствуясь материалами предыдущего занятия

**см. сцены 3 – 7*

4. Практический блок.

Векторная графика

Работа с модулем turtle подразумевает в большей степени выполнение практических графических проектов, поэтому освоение нового материала организовано в формате выполнения и разбора заданий с теоретическими вставками для объяснения основных понятий.

Для организации **практической работы** ученики занимают рабочие места и запускают Python (терминал IDLE) на своих компьютерах. Для выполнения практической работы используются материалы видеоролика:

- Подключение модуля turtle
- Изменение размера и цвета экрана
- Организация случайного фона
- Инструмент Перо

- Поворот и движение вперед для инструмента Перо

После демонстрации каждого задания запускается таймер. Время работы таймера определяется сложностью задания. До завершения работы таймера ученики выполняют задания на компьютерах.

После завершения работы таймера демонстрируется разбор задания. Ученики останавливают работу и обсуждают разбор задания.

**см. сцены 11 – 23 (кроме сцен с теорией)*

Практические задания разделены **теоретическими вставками**, необходимыми для работы над проектами урока:

- Модуль turtle, подключение
- Команда bgcolor
- Цветовая модель RGB
- Объект и команды объекта

**см. сцены 8 – 10, 13, 18-19*

По итогам работы ученики получают объекты, созданные с помощью векторной графики.

При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу; если ответы на вопросы вызывают у учеников затруднения, преподаватель может вывести нужную сцену ролика на экран для помощи ученикам.

5. Рефлексия. Сообщение домашнего задания.

Завершаем демонстрацией ролика и кратким обобщением материалов занятия. Преподаватель возвращается к зафиксированному в ходе дискуссии в начале урока предположениям учеников и обсуждает насколько их предположения были правильными, делаются выводы.

Преподаватель дает ученикам домашнее задание к следующему занятию (*Приложение 2*).

**см. сцена 24.*

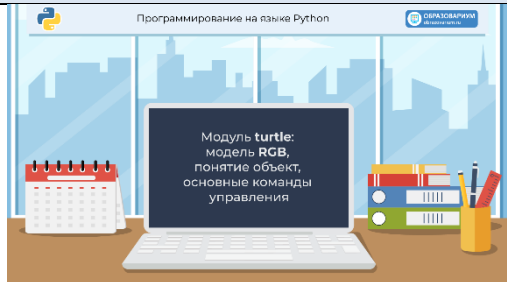
Сценарный план видеоролика

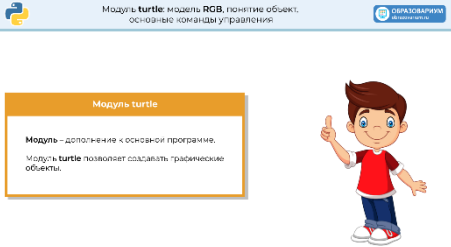
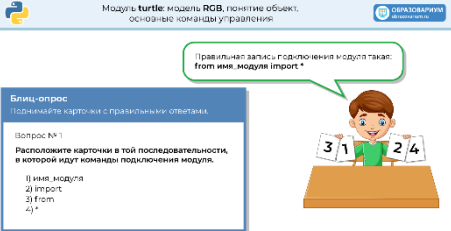
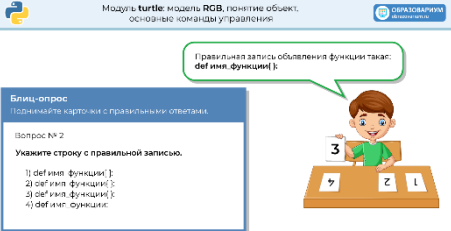
В таблице «Содержание видеоролика» представлены:

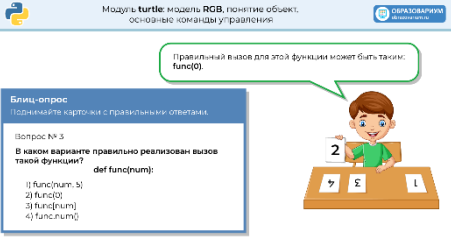
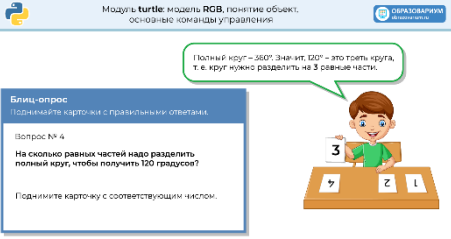
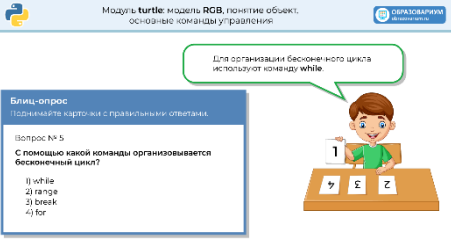
- название блоков видеоролика (тайминг);
- краткое описание содержания в каждом блоке;
- фрагменты из видеоролика, относящиеся к соответствующему блоку;
- номера сцен в каждом блоке.

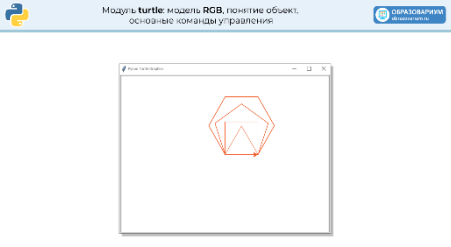
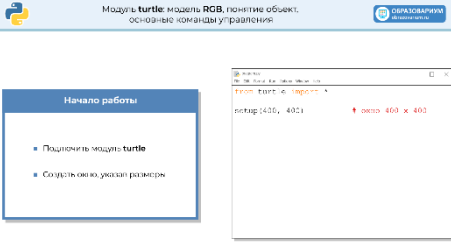
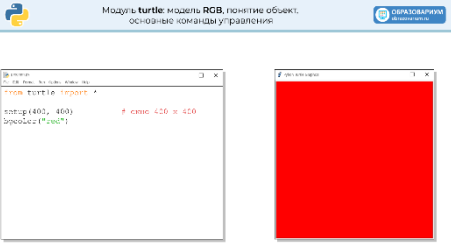
Учитель при подготовке к уроку может ознакомиться с содержанием видеоролика в текстовом формате, при необходимости распечатать фрагменты текста или примеры заданий и задач для использования в работе с учениками. Распечатанные тексты и задания из таблицы также можно применять в качестве раздаточного материала как на уроке, так и для домашних заданий.

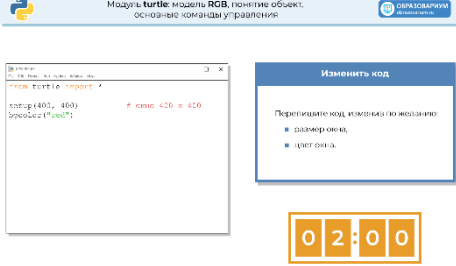

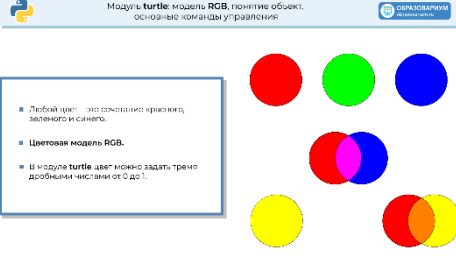
Таблица. Содержание видеоролика

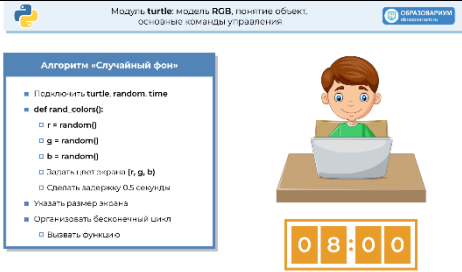
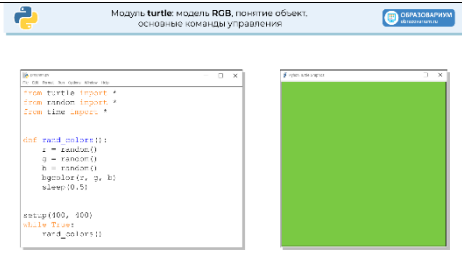
Название блока	Содержание блока и комментарии	Фрагменты из видеоролика	№ сцен
Вводный блок. Мы узнаем	Обозначаем ученикам тему и цели урока. Модуль turtle: модель RGB, понятие объект, основные команды управления	 Сцена 1	1 2

	<p>Из прошлых уроков нам известно, что модуль – это дополнение к основной программе. Модули содержат дополнительные функции и расширяют функционал программы.</p> <p>Мы начинаем знакомство с модулем turtle, который поможет нам изучить основы создания изображений на экране, при помощи так называемой векторной графики.</p>	 <p>Модуль turtle – модель RGB, понятие объект, основные команды управления</p> <p>Модуль turtle</p> <p>Модуль – дополнение к основной программе. Модуль turtle позволяет создавать графические объекты.</p> <p>Сцена 2</p>	
Блок повторения.	<p><i>Повторение материала предыдущего урока; на столе имеются пронумерованные карточки; после каждого вопроса выбираем ту, номер которой, совпадает с правильным ответом.</i></p> <p>Первый вопрос. Вопрос № 1</p> <p>Расположите карточки в той последовательности, в которой идут команды подключения модуля</p> <ol style="list-style-type: none"> 1) имя_модуля 2) import 3) from 4) * <p><i>Правильный ответ: 3-1-2-4. from имя_модуля import *</i></p>	 <p>Модуль turtle – модель RGB, понятие объект, основные команды управления</p> <p>Блиц-опрос</p> <p>Подберите карточки с правильными ответами.</p> <p>Вопрос № 1</p> <p>Расположите карточки в той последовательности, в которой идут команды подключения модуля.</p> <ol style="list-style-type: none"> 1) имя_модуля 2) import 3) from 4) * <p>Правильная запись подключения модуля такая: from имя_модуля import *</p> <p>Сцена 3</p>	3 4 5 6 7
	<p>Второй вопрос. Укажите строку с правильной записью.</p> <ol style="list-style-type: none"> 1) def имя_функции[]: 2) def имя_функции{ }: 3) def имя_функции(): 4) def имя_функции: <p><i>Ответ 3. Функция создается командой def и имеет круглые скобки.</i></p>	 <p>Модуль turtle – модель RGB, понятие объект, основные команды управления</p> <p>Блиц-опрос</p> <p>Подберите карточки с правильными ответами.</p> <p>Вопрос № 2</p> <p>Укажите строку с правильной записью.</p> <ol style="list-style-type: none"> 1) def имя_функции[]: 2) def имя_функции{ }: 3) def имя_функции(): 4) def имя_функции: <p>Правильная запись объявления функции такая: def имя_функции():</p> <p>Сцена 4</p>	



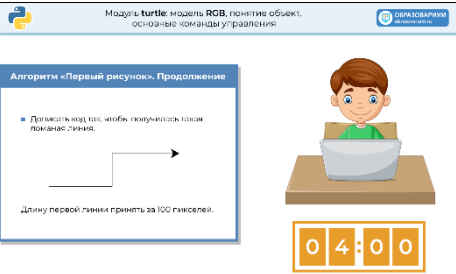
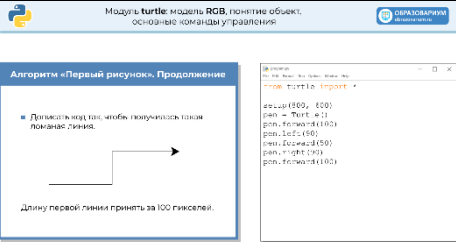
	<p>Третий вопрос. В каком варианте правильно реализован вызов такой функции?</p> <pre>def func(num):</pre> <ol style="list-style-type: none"> 1) <code>func(num, 5)</code> 2) <code>func(0)</code> 3) <code>func[num]</code> 4) <code>func.num()</code> <p><i>Ответ 2. Функция создана с одним аргументом, значит при ее вызове в круглых скобках должна быть одна ячейка с данными.</i></p>	 <p>Модуль turtle: модель RGB, понятие объект, основные команды управления</p> <p>Правильный вызов для этой функции может быть таким: <code>func(0)</code></p> <p>Блиц-опрос: Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 3: В каком варианте правильно реализован вызов такой функции?</p> <p>1) <code>func(num, 5)</code> 2) <code>func(0)</code> 3) <code>func[num]</code> 4) <code>func.num()</code></p> <p>def func(num):</p> <p>Сцена 5</p>	
	<p>Четвертый вопрос. На сколько равных частей надо разделить полный круг, чтобы получить 120 градусов?</p> <p>Поднимите карточку с соответствующим числом.</p> <p><i>Ответ 3. Полный круг составляет 360 градусов. Значит 120 — это одна треть.</i></p>	 <p>Модуль turtle: модель RGB, понятие объект, основные команды управления</p> <p>Полный круг – 360°. Значит, 120° – это треть круга, т.е. круг нужно разделить на 3 равные части.</p> <p>Блиц-опрос: Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 4: На сколько равных частей надо разделить полный круг, чтобы получить 120 градусов?</p> <p>Поднимите карточку с соответствующим числом.</p> <p>Сцена 6</p>	
	<p>Пятый вопрос. С помощью какой команды организовывается бесконечный цикл?</p> <ol style="list-style-type: none"> 1) <code>while</code> 2) <code>range</code> 3) <code>break</code> 4) <code>for</code> <p><i>Ответ 1 Нам не обойтись без команды while</i></p>	 <p>Модуль turtle: модель RGB, понятие объект, основные команды управления</p> <p>Для организации бесконечного цикла используют команду <code>while</code>.</p> <p>Блиц-опрос: Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 5: С помощью какой команды организовывается бесконечный цикл?</p> <p>1) <code>while</code> 2) <code>range</code> 3) <code>break</code> 4) <code>for</code></p> <p>Сцена 7</p>	

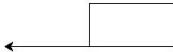




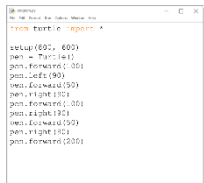
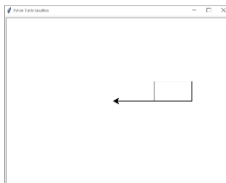

<p>Теоретический блок.</p> <p>Векторная графика</p>	<p>С помощью модуля turtle можно программировать векторную графику, то есть составлять рисунки из линий, точек и кривых.</p> <p>Чтобы начать работу нам необходимо данный модуль подключить стандартной командой import.</p> <p>Создание изображений будет происходить в отдельном окне, которое нам надо создать, указав для этого его размеры по горизонтали и вертикали.</p> <p>Для этого используется команда setup.</p> <p>Обратите внимание, что данные надо указать через запятую.</p> <p>Это и будет наш холст для рисования.</p>	 <p>Сцена 8</p>	<p>8 9 10</p>
	<p>Но сначала давайте его перекрасим.</p> <p>Команда bgcolor изменит цвет нашего экрана.</p> <p>Bg – это сокращенно «бэкграунд», то есть задний план.</p> <p>А цвет указывается в кавычках с использованием английских слов.</p>	 <p>Сцена 9</p>	
	<p>bgcolor("red")</p> <p>Эта строка перекрасит наше окно в красный цвет.</p>	 <p>Сцена 10</p>	

<p>Блок заданий. Практические задания: Задание 1.</p>	<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p>Задание 1. Алгоритм «Цвет».</p> <p>Перепишите код, изменив по желанию размер окна и цвет.</p>	 <p>Модуль turtle: модель RGB, понятие объект, основные команды управления</p> <p>Изменить код</p> <p>Перепишите код, изменив по желанию:</p> <ul style="list-style-type: none"> ■ размер окна, ■ цвет фона. <p>02:00</p> <p>Сцена 11</p>	<p>11 12</p>
<p>Теоретический блок</p> <p>Цветовая модель</p>	<p>Для компьютера любой цвет это сочетание красного, зеленого и синего. Это называется цветовой моделью RGB. Так отображает цвета ваш монитор: разные пропорции красного, зеленого и синего позволяют получить остальные цвета (как смешение синей и красной краски дает фиолетовую или как из желтой и красной выходит оранжевая)</p> <p>То есть цвет – это три числа определенного диапазона. В turtle это должны быть дробные числа от нуля до единицы.</p>	 <p>Модуль turtle: модель RGB, понятие объект, основные команды управления</p> <p>Сцена 12</p>	<p>13</p>
		 <p>Модуль turtle: модель RGB, понятие объект, основные команды управления</p> <p>Сцена 13</p>	

<p>Блок заданий. Практические задания: Задание 2.</p>	<p><i>После окончания дикторского текста запускается таймер на 8 мин.</i></p> <p>Задание 2. Алгоритм «Случайный фон»</p> <ul style="list-style-type: none"> • Подключить random, time, turtle • def rand_colors (): <ul style="list-style-type: none"> ○ r = random () ○ g = random () ○ b = random () ○ Задать цвет экрана (r, g, b) ○ Сделать задержку 0.5 секунды • Указать размер экрана • Организовать бесконечный цикл <p>Вызвать функцию</p>	 <p>Сцена 14</p>
	<p>Разбор задания 2. Код программы будет выглядеть так:</p> <pre> from turtle import * from random import * from time import * def rand_colors(): r = random() g = random() b = random() bgcolor(r, g, b) sleep(0.5) setup(400, 400) while True: rand_colors() </pre>	 <p>Сцена 15</p>

<p>Блок заданий. Практические задания: Задание 3.</p>	<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p>Задание 3. Алгоритм «Первый рисунок»</p> <ul style="list-style-type: none"> Подключить turtle Указать размер экрана 800 x 600 <p>pen = Turtle()</p>	<div data-bbox="1480 228 1986 499"> </div> <p>Сцена 16</p>	<p>16 17</p>
<p>Теоретический блок</p> <p>Понятие объекта</p>	<p>Прозвучало слово объект. Что это такое?</p> <p>Объект – это некое техническое средство, обладающее методами и свойствами, способное выполнять заданный алгоритм.</p> <p>Объект, который мы создали умеет двигаться, поворачиваться, перемещаться в заданную точку, скрываться, показываться. Он может менять свой размер, форму, толщину рисуемой линии, ее цвет и много еще чего.</p> <p>Все эти умения дает ему модуль turtle, чьим экземпляром он является.</p>	<div data-bbox="1480 903 1986 1158"> </div> <p>Сцена 18</p>	<p>18 19</p>

<p>Теоретический блок</p> <p>Команды объекта</p>	<p>С помощью этого пера мы и будем рисовать. Для этого необходимо написать имя объекта, поставить точку и прописать команду, которую надо выполнить.</p> <p>Начнем с самых простых.</p> <p>forward – идти вперед на указанное количество пикселей.</p> <p>Пиксели – точки, из которых состоит наш экран</p> <p>Команды left или right – повернуть стрелку на указанное количество градусов влево или вправо.</p>	 <p>Сцена 19</p>	
<p>Блок заданий.</p> <p>Практические задания:</p> <p>Задание 3. продолжение</p>	<p><i>После окончания дикторского текста запускается таймер на 4 мин.</i></p> <p>Задание 3. Алгоритм «Первый рисунок». Продолжение</p> <ul style="list-style-type: none"> Дописать код так, чтобы получилась такая ломаная линия  <p>Длину первой линии принять за 100 пикселей.</p>	 <p>Сцена 20</p>	<p>20 21</p>
	<p>Разбор задания 3. Код программы будет выглядеть так:</p> <p><i>Код дописывается</i></p> <pre>pen.forward(100) pen.left(90) pen.forward(50) pen.right(90) pen.forward(100)</pre>	 <p>Сцена 21</p>	

<p>Блок заданий. Практические задания: Задание 3. окончание</p>	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p>Задание 3. Алгоритм «Первый рисунок». Окончание</p> <ul style="list-style-type: none"> Дописать код так, чтобы стрелка вернулась на место старта 	<div> <div>  <div> <div>Модуль turtle: модель RGB, понятие объект, основные команды управления</div> <div>описывающие</div> </div> </div> <div> <div>Алгоритм «Первый рисунок». Продолжение</div> <ul style="list-style-type: none"> Дополнить код так, чтобы стрелка вернулась на место старта.  </div> <div>  <div>03:00</div> </div> </div> <p>Сцена 22</p>	<p>22 23</p>
	<p>Разбор задания 3. Код программы будет выглядеть так: <i>Код дописывается</i></p> <pre> pen.right(90) pen.forward(50) pen.right(90) pen.forward(200) </pre>	<div> <div>  <div> <div>Модуль turtle: модель RGB, понятие объект, основные команды управления</div> <div>описывающие</div> </div> </div> <div>   </div> </div> <p>Сцена 23</p>	
<p>Блок завершения занятия. Рефлексия. Сообщение домашнего задания</p>	<p><i>Завершаем демонстрацией ролика и кратким обобщением материалов занятия.</i></p> <p>Подведем итоги.</p> <p>Мы узнали:</p> <ul style="list-style-type: none"> Как с помощью модуля turtle нарисовали несколько линий, используя повороты влево и вправо, а также движение вперед. познакомились с цветовой моделью RGB. <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	<div> <div>  <div> <div>Модуль turtle: модель RGB, понятие объект, основные команды управления</div> <div>описывающие</div> </div> </div> <div> <div>Сегодня на уроке</div> <ul style="list-style-type: none"> С помощью модуля turtle нарисовали несколько линий, используя повороты влево и вправо, а также движение вперед. Познакомились с цветовой моделью RGB. </div> <div> <div>Домашнее задание</div> <p>ДЗ</p> <p>Найдите в сети Интернет информацию о цветовой модели RGB и укажите, каковы пропорции и соотношения для создания основных цветов.</p> <p>Подготовьте: это может быть задание по астрономии и браузерная таблица цветов HTML.</p> </div> </div> <p>Сцена 24</p>	<p>24</p>

Домашнее задание

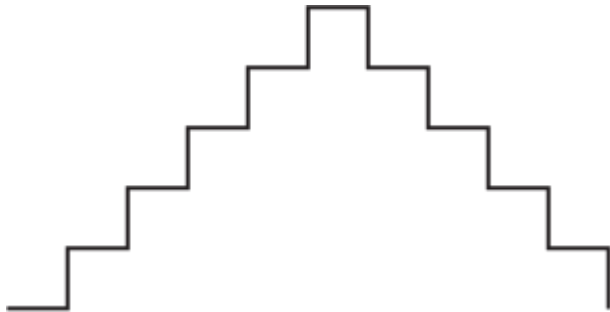
Найдите в различных источниках (интернет, литература) информацию о цветовой модели RGB и посмотрите какие пропорции необходимы для создания основных цветов. Это можно сделать по запросу в браузере «таблица цветов HTML».

Задание можно выполнить на компьютере и представить результат и код в виде файла или снимка экрана, или распечатки.

Практика

Проект «Лесенка»

С помощью модуля turtle создайте такой рисунок. На свое усмотрение задайте размер экрана и цвет фона.



Используйте два конечных цикла: один для восходящего движения, другой – для нисходящего.

Для команд forward, right, left можно использовать сокращенную запись.

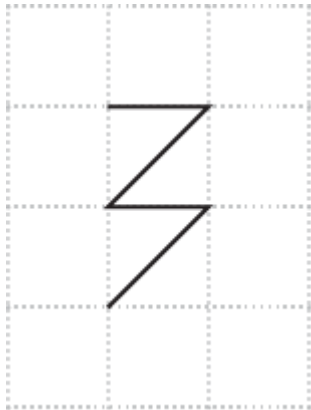
```
program.py
File Edit Format Run Options Window Help

from turtle import *

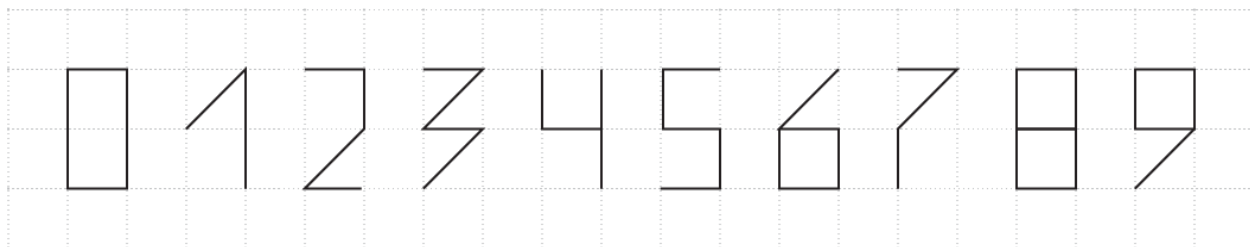
pen = Turtle()
pen.fd(10)  # сокращенная запись для pen.forward(10)
pen.lt(90)  # сокращенная запись для pen.left(90)
pen.rt(90)  # сокращенная запись для pen.right(100)
```

Проект «Цифры»

С помощью модуля turtle нарисуйте цифру 3:



В качестве тренировки вы можете нарисовать остальные цифры.



Приложение 3

Краткие организационно-методические рекомендации по организации работы на занятии

«Модуль turtle: модель RGB, понятие объект, основные команды управления».

Мы начинаем знакомство с новым модулем, который, во-первых, поможет нам узнать, как в Python можно работать с графикой, во-вторых, позволит еще раз на практике применить такие инструменты как циклы, функции, списки, рандом. Повтору именно этих тем можно посвятить вступительную часть урока, а также основами геометрии: углы и градусы.

Перед просмотром блока повторения из ролика необходимо раздать дидактический материал для выполнения заданий (по 4 пронумерованных карточки)

Во время голосований карточками можно останавливать ролик и вести учет правильных ответов.

По окончании блока – отметить тех, у кого наилучший результат. Далее карточки необходимо собрать.

В ходе урока будет создано два проекта. Один посвящен работе с цветом, и вы можете более подробно рассказать о комбинировании цветов, взяв за основу работу художников с палитрой. Также можно показать таблицу цветов с указанием пропорций красного/зеленого/синего. Также можно для наглядности вставить в функцию команду принт с выводом на экран значений переменных. Общая идея – чем ближе числа к 0, тем темнее цвет.

Во время второго проекта необходимо убедиться, что всем знакомы такое понятие как «прямой угол» и чему он равен.

Имя объекта можно выбирать самостоятельно. Главное, чтобы оно не было длинным, иначе возможны ошибки при составлении кода. Необходимо заострить внимание, что для выполнения команды – необходимы данные: градусы для поворота, пиксели – для движения.

Если останется время, то можно разнообразить код своими идеями и не обязательно с прямыми углами.