



Методические рекомендации по теме «Округление чисел: функция **round** и модуль **math**»

Цель:

- применение функций для округления чисел и решение прикладных задач с округлением чисел на языке Python.

Задачи:

- расширение кругозора обучающихся в области информатики и программирования;
- знакомство с округлением чисел на языке Python;
- решение программных прикладных задач с округлением чисел на языке Python;
- ранняя профориентация школьников, профессиональная деятельность программиста;
- развитие интеллектуальных способностей, логического и критического мышления.

Планируемые результаты

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

Предметные: обучающиеся получают представления: о возможностях функции `round` на языке Python; об основных операциях с округлением чисел на языке программирования «Python»; о прикладном использовании операций с округлением чисел в программных проектах; о возможностях и особенностях применения округления чисел в практике работы программиста.

Метапредметные: обучающиеся получают возможность владения обще предметными понятием «срез», «индекс»; владение информационно-логическими умениями; владение умениями самостоятельно планировать пути достижения целей; умениями принятия решений и осуществления осознанного выбора; повысят уровень ИКТ-компетентности.

Материалы к занятию

Приложение 1: Сценарный план видеоролика

Приложение 2: Домашние 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 (мы узнаем, про расширенные возможности функции округления числа (**round**) на языке Python; познакомимся с математическим модулем **math** и поймем для чего этот модуль может пригодиться программисту).

Проблемная дискуссия по вопросам:

- Что такое округление числа?
- Какие виды округления чисел вам известны?
- По каким правилам производится округление числа?
- Приведите пример, когда точное большое количество знаков после запятой дроби не имеет существенного значения? (например, тысячные доли копеек при финансовых вычислениях).

- Для чего, по вашему мнению, округление чисел необходимо программисту в работе? Приведите примеры.

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

- округление чисел нужно для оптимизации и упрощения результатов вычисления, когда после действий получается дробное число с большим количеством знаков после запятой;
- числа округляются по определенным правилам (напомнить детям правила округления).

Преподаватель называет ученикам тему и цели урока.

2. Вводный блок.

Тема.

Преподаватель при необходимости останавливая трансляцию, комментируя дополнительно тему занятия.

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 **в сцене 3**; 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. Теоретический блок.

Округление до разряда. Модуль math.

Продолжение демонстрации ролика с дальнейшим обсуждением вопросов:

- При помощи какой функции мы можем округлить число до целой части?
- Для такое модуль?
- Зачем нужен модуль Math?
- Что значит округлить число в большую или меньшую сторону?

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.

**см. сцены 8 – 13*

5. Блок заданий.

Проекты: «Округление», «Площадь шара».

К началу демонстрации блока заданий ученики должны занять рабочие места и запустить Python (терминал IDLE) на своих компьютерах. Блок включает **2 практических задания** для учеников с последующим разбором.

«Округление»: практическое задание 1 представляет собой 1 небольшой программный проект с использованием функций округления числа.

После выполнения задания ученики получают работающий программный продукт – программа запрашивает число и с помощью функций округления выдает полученный результат.

«Площадь шара»: практическое задание 2 представляет собой 1 небольшой программный проекта с использованием функций округления числа.

После выполнения заданий ученики получают работающий программный продукт – программа для вычисления площади шара и вывода результата (с использованием функции округления).

Блок включает в себя теоретические вставки: «Модуль **math**», «Факториал».

На сцене разбора задания преподаватель ставит ролик на паузу и вместе с учениками проводит разбор задания.

**см. сцены 14 – 26*

6. Рефлексия. Сообщение домашнего задания.

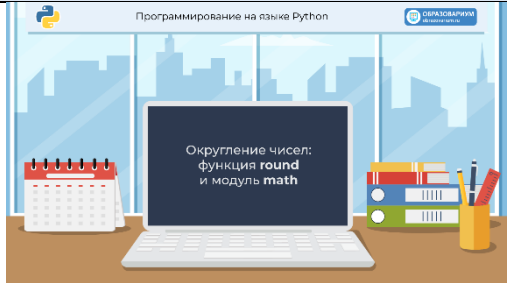
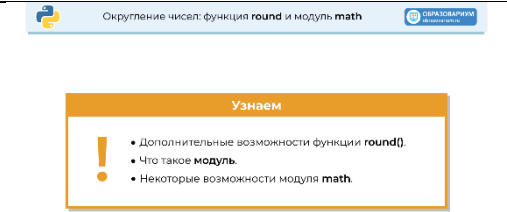
Завершаем демонстрацией ролика и кратким обобщением материалов занятия. Преподаватель возвращается к зафиксированным в ходе дискуссии в начале урока предположениям учеников и обсуждает насколько их предположения были правильными, делаются выводы.




Преподаватель дает ученикам домашнее задание к следующему занятию (*Приложение 2*). **см. сцена 27*

Сценарный план видеоролика







В таблице «Содержание видеоролика» представлен краткий текст из видеоролика, примеры заданий и задач, которые будут демонстрироваться на экране. 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.

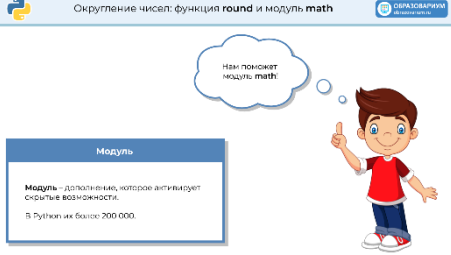
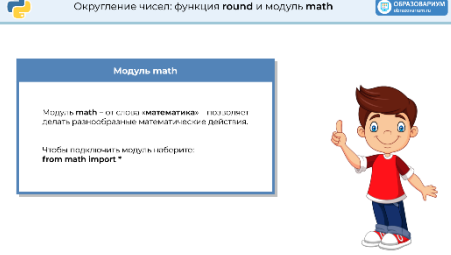
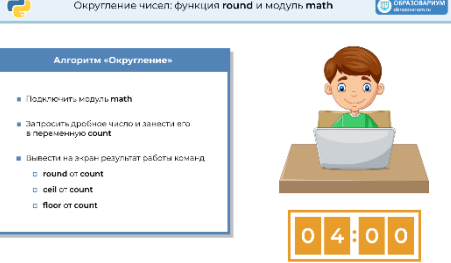
Таблица. Содержание видеоролика

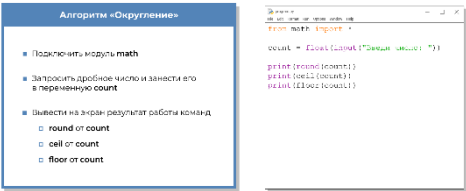
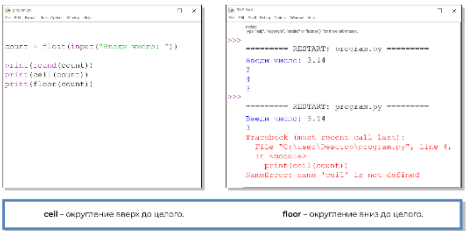
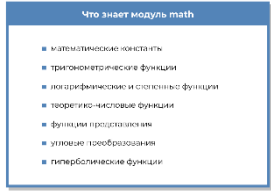
Название блока	Содержание блока и комментарии	Фрагменты из видеоролика	№ сцен
Вводный блок. Мы узнаем	Обозначаем ученикам тему и цели урока. Округление чисел: функция round и модуль math .	 Сцена 1	1 2
	Узнаем <ul style="list-style-type: none"> дополнительные возможности функции round() что такое модуль некоторые возможности модуля math 	 Сцена 2	

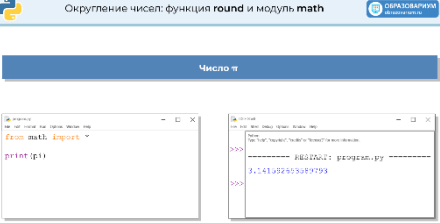
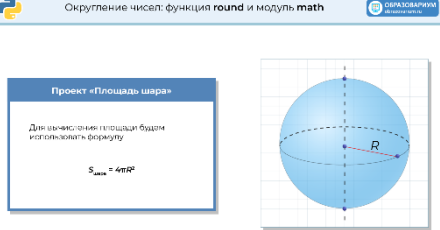
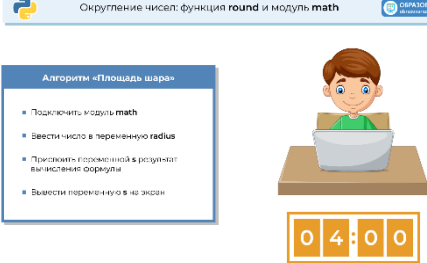
<p>Блок повторения.</p> <p>Блиц-опрос</p>	<p><i>Повторение материала предыдущего урока; на столе имеются пронумерованные карточки; после каждого вопроса выбираем ту, номер которой, совпадает с правильным ответом.</i></p> <p>Первый вопрос. Какой вариант содержит не правильную запись уменьшения num в 2 раза?</p> <ol style="list-style-type: none"> 1) num = num / 2 2) num = num // 2 3) num /= 2 4) num = num : 2 <p><i>Ответ 4. « : » – не отвечает за деление. num /= 2 – упрощенная запись для num = num / 2</i></p> <p>Второй вопрос. Какой цикл не сработает ни разу, если count = 1?</p> <ol style="list-style-type: none"> 1) while count == 1: 2) while "1" != count: 3) while "1" == count: 4) while count == count: <p><i>Ответ 3. while "1" == count: не сработает, поскольку текст не может быть равен числу.</i></p> <p>Третий вопрос. Каким методом преобразовать "текст" в "Текст"?</p> <ol style="list-style-type: none"> 1) upper () 2) lower () 3) title () 4) никаким <p><i>Ответ 4. Первые три варианта не выполняют такого преобразования.</i></p>	<div data-bbox="1509 228 1960 263"> <p>Округление чисел: функция round и модуль math</p> <p>ОЦЕКАВАРИУМ</p> </div> <div data-bbox="1509 279 1960 470"> <p>« : » не отвечает за деление. num /= 2 – упрощенная запись для num = num / 2</p> <p>Блиц-опрос Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 1 Какой вариант содержит неправильную запись уменьшения num в 2 раза?</p> <ol style="list-style-type: none"> 1) num = num / 2 2) num = num // 2 3) num /= 2 4) num = num : 2  </div> <p>Сцена 3</p> <div data-bbox="1509 622 1960 657"> <p>Округление чисел: функция round и модуль math</p> <p>ОЦЕКАВАРИУМ</p> </div> <div data-bbox="1509 673 1960 865"> <p>while "1" == count: не сработает, поскольку текст не может быть равен числу.</p> <p>Блиц-опрос Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 2 Какой цикл не сработает ни разу, если count = 1?</p> <ol style="list-style-type: none"> 1) while count == 1: 2) while "1" != count: 3) while "1" == count: 4) while count == count:  </div> <p>Сцена 4</p> <div data-bbox="1509 933 1960 968"> <p>Округление чисел: функция round и модуль math</p> <p>ОЦЕКАВАРИУМ</p> </div> <div data-bbox="1509 984 1960 1176"> <p>Первые три варианта не выполнят такого преобразования.</p> <p>Блиц-опрос Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 3 Каким методом преобразовать "текст" в "Текст"?</p> <ol style="list-style-type: none"> 1) upper() 2) lower() 3) title() 4) никаким  </div> <p>Сцена 5</p>	<p>3</p> <p>4</p> <p>5</p> <p>6</p> <p>7</p>
--	---	--	--

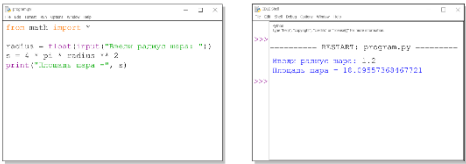
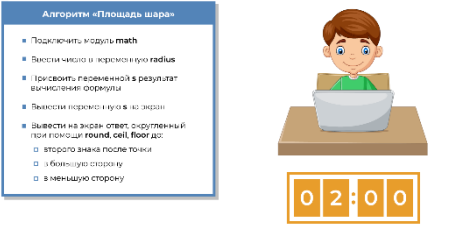
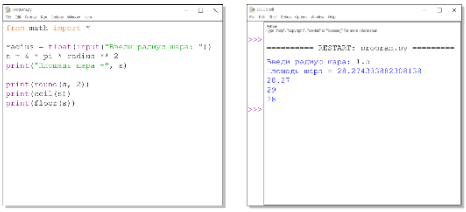
	<p>Четвертый вопрос. Какого индекса не может быть в слове "компьютер"?</p> <p>1) -9 2) 0 3) 9 4) все имеются</p> <p>Ответ 3. Индекса 9 быть не может. В этом слове положительные индексы будут от 0 до 8, а отрицательные от -1 до -9.</p>	<div><div><div>Округление чисел: функция round и модуль math</div><div><div>Блиц-опрос</div><div>Поднимайте карточки с правильными ответами.</div><div>Вопрос № 4</div><div>Какого индекса не может быть в слове "компьютер"?</div><div>1) 9 2) 0 3) 9 4) все имеются</div></div><div><div>Индекс 9 быть не может. В этом слове положительные индексы будут от 0 до 8, а отрицательные от -1 до -9.</div><div><div>3</div><div>7 2 1</div></div></div></div><div>Сцена 6</div></div>	
	<p>Пятый вопрос. Выберите правильный вариант среза "форма" из слова "информатика".</p> <p>1) [3:9] 2) [2:7] 3) [2:6] 4) [3:8]</p> <p>Ответ 2. Отсчет срезов начинается с 0, поэтому первый индекс равен 2.</p>	<div><div><div>Округление чисел: функция round и модуль math</div><div><div>Блиц-опрос</div><div>Поднимайте карточки с правильными ответами.</div><div>Вопрос № 5</div><div>Выберите правильный вариант среза "форма" из слова "информатика".</div><div>1) [5:9] 2) [2:7] 3) [2:6] 4) [3:8]</div></div><div><div>"информатика"</div><div>"форма" == "информатика"[2:7]</div><div><div>2</div><div>7 5 1</div></div></div></div><div>Сцена 7</div></div>	
<p>Теоретический блок.</p> <p>Округление до разряда.</p>	<p>При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу</p> <p>Вспомним, что при работах с формулами или другими вычислениями, ответ зачастую получается в виде дробного числа, с определенным количеством знаков после точки.</p> <p>Округлить такое число до ближайшего целого поможет функция round.</p>	<div><div><div>Округление чисел: функция round и модуль math</div><div><div>$t_1 = 10 \text{ мин}$ $S_1 = 900 \text{ м}$</div><div>$t_2 = 15 \text{ мин}$ $S_2 = 14 \text{ км}$</div><div>$t_3 = 10 \text{ мин} = 10/60 \text{ ч}$ $S_1 = 900 \text{ м} = 900/1000 \text{ км} = 0.9 \text{ км}$ $v_1 = \frac{S_1}{t_1} = \frac{0.9}{10/60} = 5.4 \text{ км/ч}$</div><div>$t_2 = 15 \text{ мин} = 15/60 \text{ ч}$ $S_2 = 14 \text{ км}$ $v_2 = \frac{S_2}{t_2} = \frac{14}{15/60} = 5.6 \text{ км/ч}$</div></div><div><div>$v_1 = ?$</div><div>$v_2 = ?$</div></div><div><div>round() – округляет до ближайшего целого числа.</div><div>round(5.4) == 5 round(5.6) == 6</div></div></div></div> <div>Сцена 8</div>	<p>8 9 10 11 12 13</p>

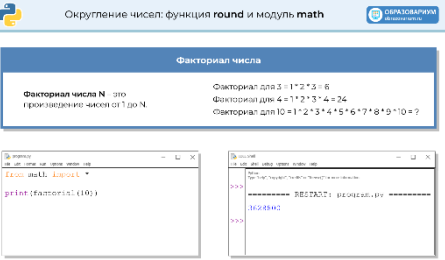
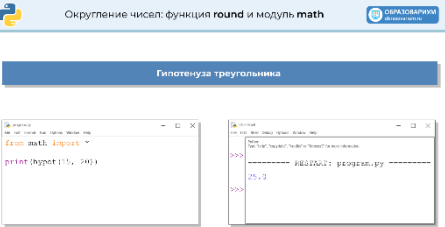
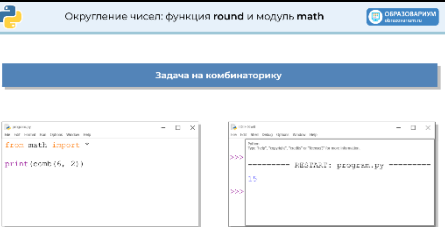
	<p>Однако не всегда результатом округления должно быть целое число. Например, при указании температуры тела используется один разряд после точки, или при указании цены в рублях – два разряда.</p>	<div data-bbox="1509 228 1962 256">  Округление чисел: функция round и модуль math  </div> <div data-bbox="1509 256 1962 496"> </div>	
	<p>Функция round умеет создавать такие форматы. Чтобы округлить число до какого-то разряда, нужно в скобках, после округляемого числа, поставить запятую и указать номер этого разряда. Если вторым параметром поставить ноль, то получится дробное число с нулем в первом разряде после точки. Вот как это выглядит. round(3.14159, 0) 3.0</p>	<div data-bbox="1509 539 1962 568">  Округление чисел: функция round и модуль math  </div> <div data-bbox="1509 568 1962 775"> </div>	
	<p>Как быть, если мы хотим округлить число не до ближайшего целого, а именно в большую или меньшую сторону? Например, 3.14 округлить до 4? Или 36.6 до 36?</p>	<div data-bbox="1509 850 1962 879">  Округление чисел: функция round и модуль math  </div> <div data-bbox="1509 879 1962 1118"> </div>	<div data-bbox="1480 1134 1581 1163"> Сцена 11 </div>



	<p>В этом нам помогут дополнительные команды модуля math.</p> <p>Модуль – это дополнения к основной программе, расширение. Своего рода патч, который активирует скрытые возможности.</p> <p>В Python их более 200 000, причем часть из них уже установлена и просто требует включения.</p>	 <p>Сцена 12</p>	
	<p>Что такое модуль math?</p> <p>Модуль math содержит наиболее применяемые математические функции и константы.</p> <p>Чтобы включить этот модуль необходимо в начале программы написать следующую строку: from math import *</p>	 <p>Сцена 13</p>	
<p>Блок заданий.</p> <p>Практические задания:</p> <p>Задание 1</p>	<p><i>После окончания дикторского текста запускается таймер на 4 мин.</i></p> <p>Задание 1. Проект «Округление»</p> <ul style="list-style-type: none"> • Подключить модуль math • Запросить дробное число и занести его в переменную count • Вывести на экран результат работы команд <ul style="list-style-type: none"> – round от count – ceil от count – floor от count 	 <p>Сцена 14</p>	<p>14 15</p>

	<p>Разбор задания 1. Разберите получившийся код.</p> <pre>from math import * count = float(input("Введи число: ")) print(round(count)) print(ceil(count)) print(floor(count))</pre> <p><i>После окончания времени провести разбор решения, анализируя код.</i></p>	 <p>Сцена 15</p>	
<p>Теоретический блок 2.</p> <p>Модуль math.</p>	<p><i>При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу</i></p> <p>Теперь уберем первую строчку и запустим программу еще раз. Программа запросит число, выведет результат работы функции round и выдаст вот такую ошибку: «имя сейл не найдено». Без подключенного модуля math, Python не узнаёт эти команды.</p> <p>Команды ceil округляет число строго вверх, а floor – строго вниз; в отличие от round они не могут округлять до разряда после точки.</p>	 <p>Сцена 16</p>	<p>16 17 18 19</p>
	<p>В приведенном списке показаны возможности модуля math. Часть этого материала вы, возможно, уже начали изучать на уроках математики. Что-то из этого предстоит узнать в старших классах. А некоторый материал – это программа вузов.</p>	 <p>Сцена 17</p>	

	<p>Мы не сможем разобрать все возможности модуля math, однако кое-чем можем воспользоваться прямо сейчас.</p> <p>Например, если у нас подключен модуль math, то автоматически появится константа π, которую не надо создавать.</p> <pre>from math import * print(pi)</pre>	 <p>Сцена 18</p>	
	<p>Используем это для разработки проекта для нахождения площади шара.</p> <p>Проект «площадь шара»</p> <p>Для вычисления площади будем использовать формулу</p> $S = 4\pi R^2$	 <p>Сцена 19</p>	
<p>Блок заданий.</p> <p>Практические задания:</p> <p>Задание 2</p>	<p><i>После окончания дикторского текста запускается таймер на 4 мин.</i></p> <p>Задание 2. Алгоритм «Площадь шара»</p> <ul style="list-style-type: none"> – Подключить модуль math – Ввести в переменную radius – число – Присвоить переменной s результат вычисления формулы – Вывести переменную s на экран 	 <p>Сцена 20</p>	<p>20 21 22 23</p>

	<p>Разбор задания 2. Код программы может выглядеть так. Обратите внимание на точность вычислений!</p> <pre>from math import * radius = float(input("Введи радиус шара: ")) s = 4 * pi * radius ** 2 print("Площадь шара =", s)</pre> <p><i>После окончания времени провести разбор решения, анализируя код.</i></p>	 <p>Скриншоты кода и его выполнения. В первом окне показан код: <code>from math import *</code>, <code>radius = float(input("Введи радиус шара: "))</code>, <code>s = 4 * pi * radius ** 2</code>, <code>print("Площадь шара =", s)</code>. Во втором окне — результат выполнения: <code>Введи радиус шара: 1.2</code> и <code>Площадь шара = 16.0957344677121</code>.</p> <p>Сцена 21</p>	
	<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p>Задание 2. Алгоритм «Площадь шара» (продолжение)</p> <ul style="list-style-type: none"> – Выведите на экран, округленный при помощи round, ceil, floor ответ до: <ul style="list-style-type: none"> - второго знака после точки - в большую сторону - в меньшую сторону 	 <p>Скриншот карточки с алгоритмом «Площадь шара». Включены пункты: подключить модуль math, ввести радиус, присвоить переменной s результат вычисления, вывести на экран. Также есть иллюстрация мальчика за компьютером и таймер на 02:00.</p> <p>Сцена 22</p>	
	<p>Разбор задания 2 (продолжение). В итоге выполнения кода получается четыре результата: точный и три округленных. Код дописан следующими строками:</p> <pre>print(round(s,2)) print(ceil(s)) print(floor(s))</pre> <p><i>После окончания времени провести разбор решения, анализируя код.</i></p>	 <p>Скриншоты кода и его выполнения. В первом окне добавлен код: <code>print(round(s, 2))</code>, <code>print(ceil(s))</code>, <code>print(floor(s))</code>. Во втором окне — результат: <code>Число радиуса шара: 1.2</code>, <code>Площадь шара = 16.0957344677121</code>, <code>26.27</code>, <code>16</code>.</p> <p>Сцена 23</p>	

<p>Теоретический блок 3.</p> <p>Факториал.</p>	<p><i>При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу</i></p> <p>Факториал числа N — это произведение чисел от 1 до N. Например, факториал числа 3 равен 6, а числа 4 = 24. Найдем факториал для 10.</p> <p>Это будет 3 628 800. Для вычисления использована специальная команда</p> <pre>from math import * print(factorial(10))</pre>	 <p>Сцена 24</p>	<p>24 25 26</p>
	<p>Можно ли узнать гипотенузу треугольника по катетам без всяких формул?</p> <p>Предположим, первый катет равен 15, а второй равен 20.</p> <p>Ответ 25.</p> <p>Для вычисления использована специальная команда</p> <pre>from math import * print(hypot(15, 20))</pre>	 <p>Сцена 25</p>	
	<p>Рассмотрим задачу на комбинаторику. Имеется 6 ребят. Сколько вариантов пар они смогут составить?</p> <p>Ответ: 15 различных пар</p> <p>Для вычисления использована специальная команда</p> <pre>from math import * print(comb(6, 2))</pre>	 <p>Сцена 26</p>	

Блок завершения занятия. Рефлексия. Сообщение домашнего задания	<p><i>Завершаем демонстрацией ролика и кратким обобщением материалов занятия.</i></p> <p>Подведем итоги:</p> <ul style="list-style-type: none"> - узнали дополнительные возможности функции round(); - о многочисленных возможностях модуля math. <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	<div data-bbox="1512 231 1960 263">  Округление чисел: функция round и модуль math  </div> <div data-bbox="1556 279 1908 486"> <div data-bbox="1556 279 1908 375"> <p>Мы узнали</p> <ul style="list-style-type: none"> • Дополнительные возможности функции round() • Многочисленные возможности модуля math. </div> <div data-bbox="1556 391 1908 486"> <p>ДЗ</p> <p>Примером кода модуля, которую вы бы добавили в модуль math, если бы были разработчиком, что бы она вычисляла? Сколько бы она просила параметров и в каком виде выводила бы результат?</p> </div> </div> <div data-bbox="1482 534 1579 566">Сцена 27</div>	27
---	--	--	----

Приложение 2

Домашнее задание

Команду для вычисления чего вы бы добавили в модуль **math**, если бы были разработчиком. Сколько бы она просила параметров и в каком виде выводила бы результат?

Практика

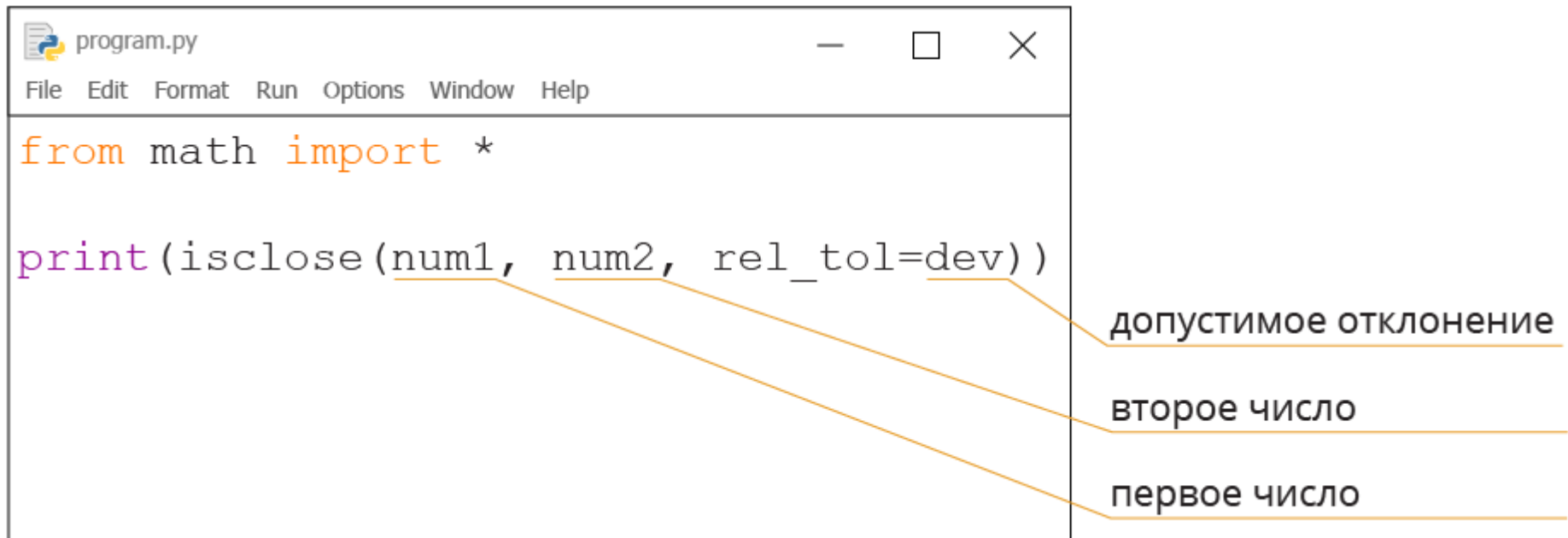
Проект «Близкие числа»

Запросите два целых числа и допустимое значение отклонения в процентах.

Будем считать числа близкими, если разница между ними меньше процента отклонения от большего числа.

Определите являются ли введенные числа близкими и выведите соответствующее сообщение.

Для определения, являются ли числа близкими, вы можете использовать метод **isclose()** из библиотеки **math**.



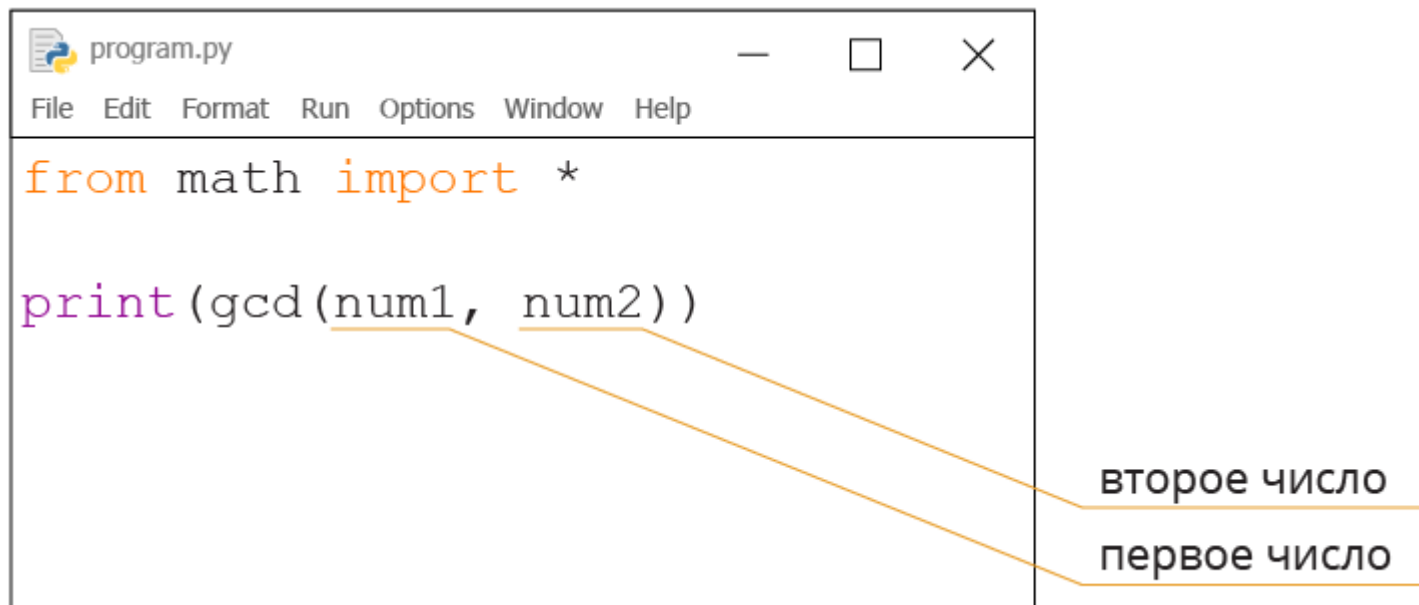
Такой код выдаст **True**, если числа близки, и **False** в обратном случае.
Допустимое отклонение задается в десятичной форме записи процентов.

Проект «Общий делитель»

Запросите два целых числа.

Выведите на экран наибольший общий делитель (**НОД**) этих чисел. Если **НОД** отсутствует, выведите соответствующее сообщение.

Для определения НОД двух чисел вы можете использовать метод **gcd()** из библиотеки **math**.



Такой код выдаст **НОД** первого и второго числа либо 1, если **НОД** отсутствует.

Приложение 3

Краткие организационно-методические рекомендации по организации работы на занятии

«Округление чисел: функция round и модуль **math**».

В начале занятия необходимо вспомнить уже известные нам методы изменения регистров текстовой информации (**upper**, **lower**, **title**), функцию, определяющую длину текста (**len**). А также терминологию слов «срез» и «индекс». Можно попрактиковаться в вычислении среза устно, на каком-нибудь слове.

Соберите у ребят домашние задания – сложные слова в виде срезов от слова «кибернетика». Если есть возможность – можно их набрать в терминале, чтобы убедиться в правильности решений.

Перед просмотром блока повторения из ролика необходимо раздать дидактический материал для выполнения заданий из блока повторение (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.

Можно сделать акцент на том, что мы не можем рассказать о всех возможностях Python на уроке. В частности, методов работы с текстами значительно больше. Поэтому любознательным в помощь интернет-порталы и учебники.

После теоретического блока необходимо убедиться, что ребятами поняли: **модули** — это не отдельные программы, это вспомогательные инструменты. А не подключены они, потому что иначе наша программа «весила» бы довольно много.

В задании 2 ребята должны самостоятельно перевести формулу с экрана на язык программирования. Константу **π** отдельно создавать не надо!

В зависимости от уровня подготовки ребят по математике, можно им рассказать про возможности работы в Python с синусами, косинусами, логарифмами, преобразованиях углов в радианы.

Словом, о том, с чем они сталкиваются на уроках математики.