



Методические рекомендации по теме «Операторы условия **if** и **else**»

Цель:

- знакомство с понятием условия на языке Python, применение для решения задач операторов **if** и **else**.

Задачи:

- расширение кругозора обучающихся в области информатики и программирования;
- знакомство с понятием условия на языке Python;
- решение программных прикладных задач с использованием операторов условия **if** и **else** 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 программировании условий на языке Python; об округлении чисел в языке программирования Python; о прикладном использовании условий в программных проектах; о возможностях и особенностях применения условных операторов в практике работы программиста.

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

Материалы к занятию

Приложение 1: Сценарный план ролика

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 (мы узнаем, как можно что такое «условие» в программировании и как работать с условиями на языке Python; мы научимся использовать условные операторы «IF» и «ELSE», сделаем несколько программных проектов, чтобы понять как условные операторы используются программистами).

Проблемная дискуссия по вопросам:

- Что такое «условие» в обычной жизни человека (приведите примеры)?
- Какими словами можно описать условие?
- Зачем нужно условие в программировании?

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

- в реальной жизни мы постоянно сталкиваемся с различными условиями;
- в программировании условие используется для написания условного алгоритма (ситуация выбора);
- условие можно описать словом (Если, то).

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.

Условный оператор.

Продолжение демонстрации ролика с дальнейшим обсуждением вопросов:

- Что такое условие в программировании?
- Какой синтаксис используется в Python для условного оператора?

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 – 10*

5. Блок заданий.

Проекты: «Положительное число», «Похвала ученику», «Максимальный рост».

К началу демонстрации блока заданий ученики должны занять рабочие места и запустить Python (терминал IDLE) на своих компьютерах.

«Положительное число»: включает *практическое задание 1* с таймером; после завершения работы таймера демонстрируется разбор задания. Задание представляет собой 2 этапа создания простейшего математического проекта с использованием условий.

После выполнения задания ученики получают работающий продукт – программу для определения положительное или отрицательное число ввел пользователь.

«Похвала ученику»: включает *практическое задание 2* с таймером; после завершения работы таймера демонстрируется разбор задания. Задание представляет собой 2 этапа создания простого логического проекта с использованием «условия».

После выполнения задания ученики получают работающий продукт – программу для организаций простого диалога с пользователем.

«Максимальный рост»: включает *практическое задание 3* с таймером; после завершения работы таймера демонстрируется разбор задания. Задание представляет собой этапы создания простейшего математического проекта с использованием «условий».

После выполнения задания ученики получают работающий продукт – программу, которая определяет являются ли два введенных числа одинаковыми или разными, в случае разных чисел выводится на экран то из них, которое больше (максимальный рост).

Блок заданий включает теоретические вставки «Что можно сравнивать», «Команда **else**», «Важность точного условия».

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

**см. сцены 11 – 25*

5. Рефлексия. Сообщение домашнего задания (4 мин).

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.

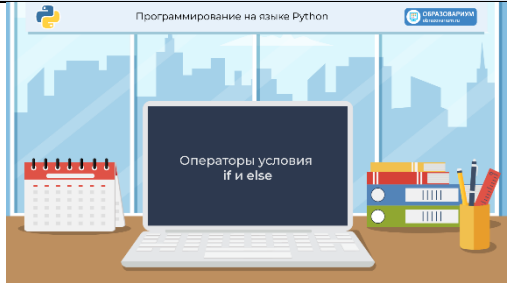
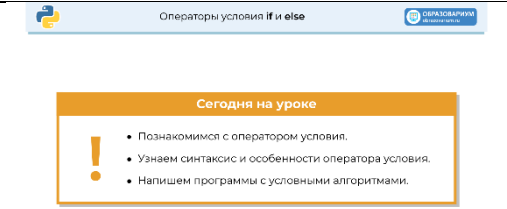
Преподаватель дает ученикам домашнее задание к следующему занятию (*Приложение 2*).

**см. сцену 26*

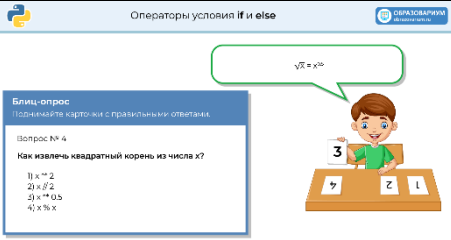
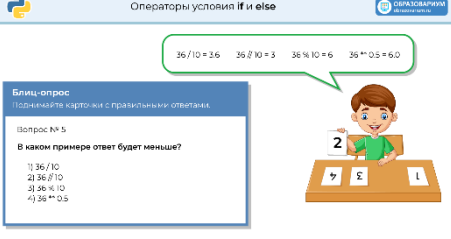

Сценарный план видеоролика

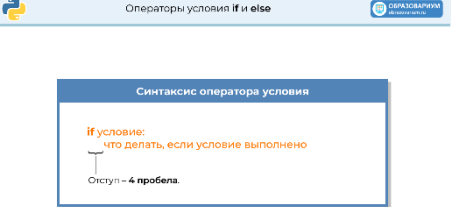
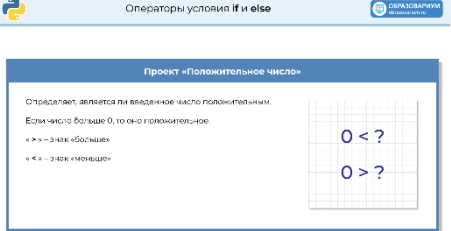
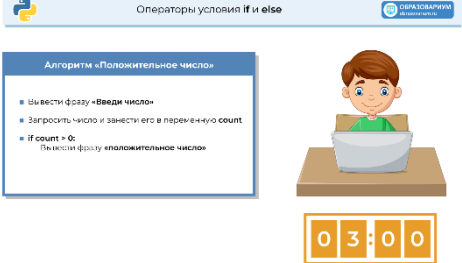
В таблице «Содержание видеоролика» представлен дикторский текст из видеоролика, примеры заданий и задач, которые будут демонстрироваться на экране. Учитель при подготовке к уроку может ознакомиться с содержанием видеоролика в текстовом формате, при необходимости распечатать фрагменты текста или примеры 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.

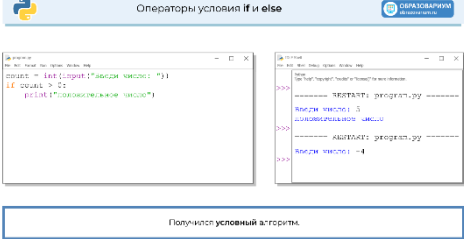
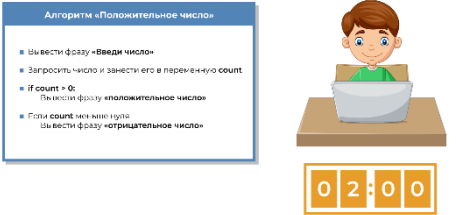
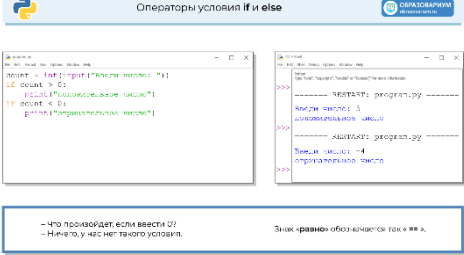
Таблица. Содержание видеоролика

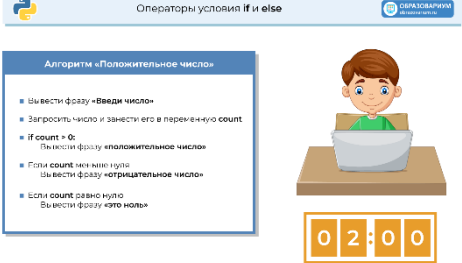
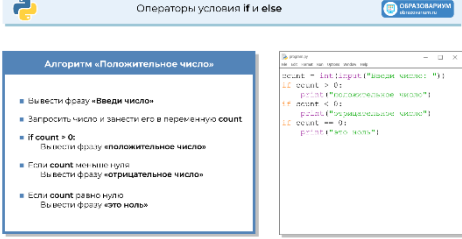
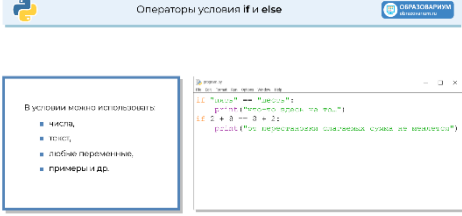
Название блока	Содержание блока и комментарии	Фрагменты из видеоролика	№ сцен
Вводный блок. Мы узнаем	<p>Обозначаем ученикам тему и цели урока.</p> <p>Операторы условия if и else</p>	 <p>Сцена 1</p>	1 2
	<p>На этом занятии:</p> <ul style="list-style-type: none"> • Познакомимся с оператором условия. • Узнаем синтаксис и особенности оператора условия. • Напишем программы с условными алгоритмами. 	 <p>Сцена 2</p>	

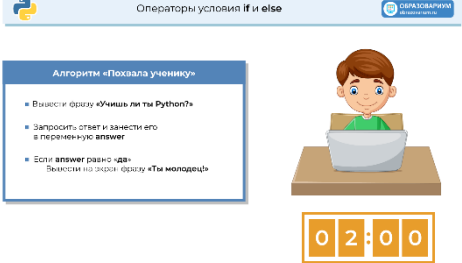
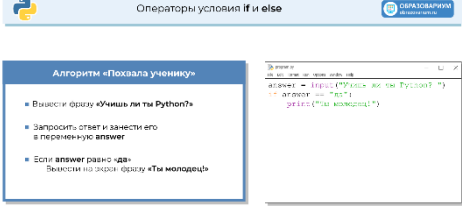
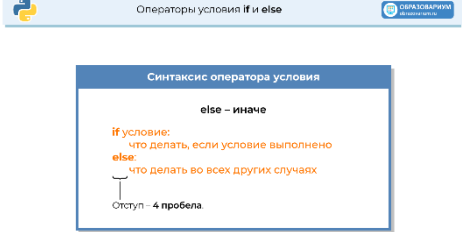
<p>Блок повторения.</p> <p>Блиц-опрос</p>	<p><i>Повторение материала предыдущего урока; на столе имеются пронумерованные карточки; после каждого вопроса выбираем ту, номер которой, совпадает с правильным ответом.</i></p> <p>Первый вопрос. Какая функция возводит число в степень?</p> <ol style="list-style-type: none"> 1) input 2) pow 3) round 4) print <p><i>ОТВЕТ № 2: pow (a, n) – это a^n; a – это основание степени, а n – показатель степени.</i></p>	<p>Операторы условия if и else</p> <p>Блиц-опрос Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 1 Какая функция возводит число в степень?</p> <ol style="list-style-type: none"> 1) input 2) pow 3) round 4) print <p>pow(a, n) – это a^n, где a – это основание степени, n – показатель степени.</p> <p>Сцена 3</p>	<p>3</p> <p>4</p> <p>5</p> <p>6</p> <p>7</p>
	<p>Второй вопрос. Что делает функция round?</p> <ol style="list-style-type: none"> 1) округляет число 2) меняет знак числа 3) извлекает корень 4) меняет тип числа <p><i>ОТВЕТ № 1: Функция round округляет введенное в нее число до указанного параметра или до целого; round (3.1415926) = 3.</i></p>	<p>Операторы условия if и else</p> <p>Блиц-опрос Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 2 Что делает функция round?</p> <ol style="list-style-type: none"> 1) округляет число 2) меняет знак числа 3) извлекает корень 4) меняет тип числа <p>Функция round округляет введенное в нее число до указанного параметра или до целого. round(3.1415926) = 3</p> <p>Сцена 4</p>	
	<p>Третий вопрос. Какой результат получится abs (-4 ** 2) =?</p> <ol style="list-style-type: none"> 1) 8 2) 16 3) -16 4) -8 <p><i>ОТВЕТ № 2: $-4 ** 2 = -16$; abs (-16) = 16.</i></p>	<p>Операторы условия if и else</p> <p>Блиц-опрос Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 3 Какой результат получится в примере: abs(-4 ** 2) = ?</p> <ol style="list-style-type: none"> 1) 8 2) 16 3) -16 4) -8 <p>$-4 ** 2 = -16$ abs(-16) = 16</p> <p>Сцена 5</p>	

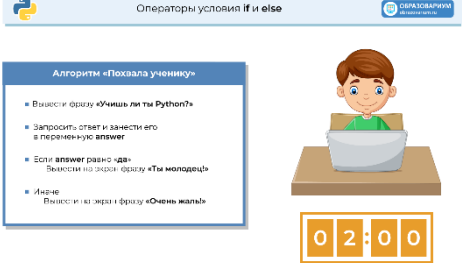
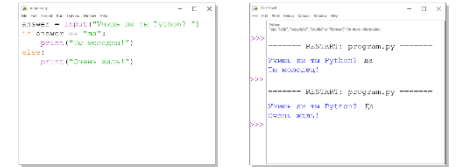
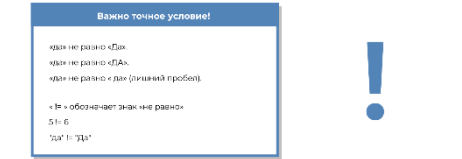
	<p>Четвертый вопрос. Как извлечь квадратный корень из числа x?</p> <ol style="list-style-type: none"> 1) $x ** 2$ 2) $x // 2$ 3) $x ** 0.5$ 4) $x \% x$ <p>ОТВЕТ № 3: $\sqrt{x} = x^{0.5}$</p>	 <p>Сцена 6</p>	
	<p>Пятый вопрос. В каком примере ответ будет меньше?</p> <ol style="list-style-type: none"> 1) $36 / 10$ 2) $36 // 10$ 3) $36 \% 10$ 4) $36 ** 0.5$ <p>ОТВЕТ № 2: $36 / 10 = 3.6$; $36 // 10 = 3$; $36 \% 10 = 6$; $36 ** 0.5 = 6.0$.</p>	 <p>Сцена 7</p>	
<p>Теоретический блок.</p> <p>Условный оператор.</p>	<p><i>При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу</i></p> <p>Существуют ситуации, в которых от ответа на вопрос зависит какое действие будет совершено. В программировании такие вопросы называют условиями. <u>Например</u>. Завтра каникулы? Если «да» – погуляю с собакой. Иначе – пойду в школу.</p> <p>Условный оператор – это команда, которая проверяет, верно ли утверждение в условии. if – если.</p>	 <p>Сцена 8</p>	<p>8 9 10</p>

	<p>Синтаксис условного.</p> <p>Сначала надо написать ключевое слово if, после него – условие, а в самом конце строки должно быть двоеточие.</p> <p>Команды, которые надо выполнить, начинаются с отступа в 4 пробела.</p>	 <p>Сцена 9</p>	
	<p>Создадим программу, которая будет определять является ли введенное число положительным.</p> <p>Положительное число – это число, которое больше нуля.</p> <p>Знаки «больше и меньше» пишутся также как и в математике.</p>	 <p>Сцена 10</p>	
<p>Блок заданий.</p> <p>Практические задания:</p> <p>Задание 1</p>	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p>Задание 1. Алгоритм «Положительное число»</p> <p>Вывести фразу «Введи число».</p> <p>Запросить число и занести его в переменную count.</p> <p>if count > 0: вывести фразу «положительное число»</p>	 <p>Сцена 11</p>	<p>11</p> <p>12</p> <p>13</p> <p>14</p> <p>15</p> <p>16</p>

<p>Разбор задания 1.</p> <p>Ваш код может выглядеть так.</p> <pre>count = int(input("Введи число: ")) if count > 0: print("положительное число")</pre> <p><i>После окончания времени провести разбор решения, анализируя код. Проведите тестирование работы программы для положительного и отрицательного числа.</i></p>	 <p>Операторы условия if и else</p> <p>Получился условный алгоритм.</p> <p>Сцена 12</p>
<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p>Задание 1. Дополним программу</p> <p>Дополните код условием: если count меньше нуля вывести фразу «отрицательное число»</p>	 <p>Операторы условия if и else</p> <p>Сцена 13</p>
<p>Ваш код может выглядеть так.</p> <pre>if count < 0: print("отрицательное число")</pre> <p><i>Рассмотрите случай, когда вводится ноль. Обсудите, что произойдет.</i></p> <ul style="list-style-type: none">– Что произойдет, если ввести 0?– Ничего, у нас нет такого условия.	 <p>Операторы условия if и else</p> <p>Сцена 14</p>

	<p><i>После окончания дикторского текста запускается таймер на 2 мин. Продолжите редактировать код программы.</i></p> <p>Допишите код:</p> <pre>если count == нулю вывести фразу «это ноль»</pre>	 <p>Операторы условия if и else</p> <p>Алгоритм «Положительное число»</p> <ul style="list-style-type: none"> Вывести фразу «Введи число» Запросить число и занести его в переменную count if count > 0: Вывести фразу «положительное число» Если count не меньше нуля Вывести фразу «отрицательное число» Если count равно нулю Вывести фразу «это ноль» 	
	<p>Дополнительные строки кода могут выглядеть так.</p> <pre>if count == 0: print("это ноль")</pre> <p><i>После окончания времени провести разбор решения, анализируя код.</i></p>	<p>Сцена 15</p>  <p>Операторы условия if и else</p> <p>Алгоритм «Положительное число»</p> <ul style="list-style-type: none"> Вывести фразу «Введи число» Запросить число и занести его в переменную count if count > 0: Вывести фразу «положительное число» Если count не меньше нуля Вывести фразу «отрицательное число» Если count равно нулю Вывести фразу «это ноль» 	
<p>Теоретический блок.</p> <p>Что можно сравнивать.</p>	<p><i>При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу</i></p> <p>В условие можно использовать числа, текст, любые переменные и даже примеры.</p> <pre>if "пять" == "шесть": print("что то здесь не то...") if 2 + 8 == 8 + 2: print("от перестановки сумма не меняется")</pre>	<p>Сцена 16</p>  <p>Операторы условия if и else</p> <p>В условии можно использовать:</p> <ul style="list-style-type: none"> числа, текст, любые переменные, примеры и др. 	<p>17</p>
		<p>Сцена 17</p>	

<p>Блок заданий. Практические задания: Задание 2</p>	<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p>Задание 2. Алгоритм «Похвала ученику»</p> <p>Вывести фразу «Учишь ли ты Python?». Запросить ответ и занести его в переменную answer. Если answer равна «да», то вывести на экран фразу «Ты молодец!».</p>	 <p>Сцена 18</p>	<p>18 19</p>
	<p>Разбор задания 2.</p> <p>Ваш код может выглядеть примерно так.</p> <pre>answer = input("Учишь ли ты Python? ") if answer == "да": print("Ты молодец!")</pre> <p><i>После окончания времени проведите разбор решения, анализируя код.</i></p>	 <p>Сцена 19</p>	
<p>Теоретический блок. Команда else.</p>	<p><i>При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу</i></p> <p>Синтаксис оператора условия</p> <pre>if условие: что делать, если условие выполнено else: что делать во всех других случаях</pre>	 <p>Сцена 20</p>	<p>20</p>

<p>Блок заданий. Практические задания: Задание 2</p>	<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p>Задание 2. Алгоритм «Похвала ученику» (дополнение)</p> <p>Дополните код строчками: <div style="margin-left: 40px;">иначе</div> <div style="margin-left: 40px;">Вывод фразы «очень жаль!»</div> </p>	<div style="text-align: right;">21 22</div>  <p>Сцена 21</p>
	<p>Разбор задания 2.</p> <p>Дополнение к коду может выглядеть примерно так.</p> <pre style="margin-left: 40px;">else: print("Очень жаль!")</pre> <p><i>После окончания времени проведите разбор решения, анализируя код. Обратите внимание, что если ввести слово «Да» с большой буквы, то срабатывает вывод ответа «Очень жаль!». Обсудите почему так происходит.</i></p>	 <p>Сцена 22</p>
<p>Теоретический блок.</p> <p>Важность точного условия</p>	<p><i>При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу</i></p> <p>Для программы важен не смысл текста – компьютер его все равно не понимает, для программы важно, чтобы было точное условие.</p> <ul style="list-style-type: none"> «да» не равно «Да». «да» не равно «ДА». «да» не равно « да» (лишний пробел). <p>Для обозначения условия «не равно» используют восклицательный знак и знак равно.</p>	<div style="text-align: right;">23</div>  <p>Сцена 23</p>

Блок заданий.
Практические задания:
Задание 3

После окончания дикторского текста запускается таймер на 6 мин.

Задание 3. Алгоритм «Максимальный рост»

Ввести число в переменную height1.
Ввести число в переменную height2.
Если переменные равны
 Вывести фразу «рост одинаковый»
Иначе
 Вывести фразу «рост разный»
Если height1 больше height2
 Вывести height1
Иначе
 Вывести height2

Разбор задания 3.

Ваш код может выглядеть примерно так.

```
count1 = int(input("Введи 1 число"))
count2 = int(input("Введи 2 число"))
if count1 == count2:
    print("Они равны")
else:
    print("они разные")
if count1 > count2:
    print(count1)
else:
    print(count2)
```

После окончания времени проведите разбор решения, анализируя результаты выполненных операций.

Операторы условия if и else

Алгоритм «Максимальный рост»

- Ввести число в переменную height1
- Ввести число в переменную height2
- Если переменные равны
 Вывести фразу «рост одинаковый»
Иначе
 Вывести фразу «рост разный»
- Если height1 больше height2
 Вывести height1
Иначе
 Вывести height2

Сцена 24

Операторы условия if и else

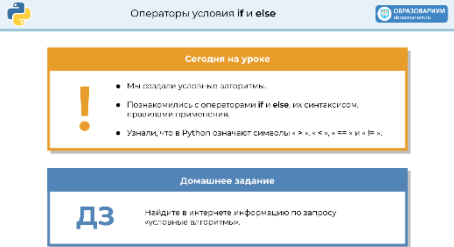
Алгоритм «Максимальный рост»

- Ввести число в переменную height1
- Ввести число в переменную height2
- Если переменные равны
 Вывести фразу «рост одинаковый»
Иначе
 Вывести фразу «рост разный»
- Если height1 больше height2
 Вывести height1
Иначе
 Вывести height2

```
count1 = int(input("Введи 1 число"))
count2 = int(input("Введи 2 число"))
if count1 == count2:
    print("Они равны")
else:
    print("они разные")
if count1 > count2:
    print(count1)
else:
    print(count2)
```

Сцена 25

24
25

<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> <ul style="list-style-type: none"> –Мы создали условные алгоритмы –Познакомились с операторами if и else, их синтаксисом, правилами применения –Узнали, что в Python означают символы «>», «<», «==» и «!=» <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	 <p>Сцена 26</p>	<p>26</p>
--	--	---	-----------

Приложение 2

Домашнее задание

Найти в любых достоверных источниках информацию об условных алгоритмах.

Практика

Проект «Возраст»

Запросите возраст пользователя.

Выведите на экран сообщение о том, какой основной документ подтверждает личность пользователя. Определите, может ли пользователь пройти обучение в автошколе, и выведите соответствующее сообщение.

Проект «Анализ числа»

Запросите целое число.

Определите, является ли число четным и делится ли оно нацело на 3. Выведите на экран соответствующие сообщения.

Краткие организационно-методические рекомендации по организации работы на занятии

«Операторы условия **if** и **else**».

В начале занятия можно вспомнить материал последнего урока: спросить какими символами изображаются известные нам операции (3 вида деления, умножение, сложение, вычитание, а также возведение в степень и извлечение квадратного корня). В чем отличие друг от друга двух последних операций при записи? (если показатель степени меньше 1, то это извлечение корня, а если больше – возведение в степень)

Проверьте как ребята выполнили домашнее задание – найти формулы, в которых присутствует возведение в степень или вычисление корня. Выберите несколько наиболее интересных и запишите их на доске, а рядом напишите, как (используя синтаксис Python) мы должны были бы их применить в программе.

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.

Перед теоретическим блоком: какие виды алгоритмов они помнят (линейный, условный, циклический). Напомнить, что представляет собой условный алгоритм (варианты событий при наличии условий) и попросить привести примеры условных алгоритмов из жизни.

Во время теоретического блока рекомендуется отдельно написать на доске или ее аналоге синтаксис применения условного оператора **if**, заострив внимание на двоеточие в конце и отступах, при написании команд действия.

Когда добавится оператор **else** – то его синтаксис надо присоединить к вышеуказанному.

Перед блоком заданий необходимо проследить, чтобы у всех был открыт компьютерный терминал для практической работы. Каждый из трех проектов должен быть написан отдельно. Обращайте внимание на название переменных – в описаниях проектов уже предложены наиболее оптимальные имена.

После каждого блока заданий, созданные проекты необходимо протестировать несколько раз с разными вводными данными. Поясняющую информацию нужно вводить непосредственно в функции **input**.

Особое внимание уделите символу «**равно**» - «**==**», вместо которого часто пишут «**=**», а также символу «**не равно**» «**!=**», который путают с «**=!**». Наряду с отсутствием двоеточия после условия и 4-х отступов перед командами действия – это самая «популярная» ошибка синтаксиса.