

Школа:		
Дата:	ФИО учителя:	
Класс:	Участвовали:	Не участвовали:
Тема урока: Условный оператор. Множественное ветвления		
Цели обучения, которые достигаются на данном уроке	Познакомить учащихся с условным оператором и множественным ветвлением применением их при составлении программ на языке программирования Python.	
Цели урока	Все учащиеся смогут: <ul style="list-style-type: none"> • уметь правильно записывать условный оператор и множественное ветвление для решения конкретных задач; Большинство учащихся смогут: <ul style="list-style-type: none"> • применять условный оператор и множественное ветвления при составлении программы для решения задач; Некоторые учащиеся смогут: <ul style="list-style-type: none"> • уметь находить ошибки в составленной программе. 	
Критерии оценивания	Учащийся: <ul style="list-style-type: none"> - поясняет принцип работы условного оператора и множественного ветвления; - исправляет ошибки в программе с условным оператором и множественного ветвления; - самостоятельно разрабатывает алгоритм и программу для решения задачи, используя условный оператор или множественное ветвления 	
Воспитание ценностей	воспитание эмоционально-положительной направленности на практическую деятельность, интереса к информатике, личной ответственности за результаты своей работы.	
Предварительные знания	Типы данных. Определение переменной. Логические выражения	
Межпредметные связи	Информатика, математика	
Запланированные этапы урока	Запланированная деятельность на уроке	Ресурсы
Начало урока ___ мин	Организационный момент. Мотивация к учебной деятельности. Целеполагание. Актуализация знаний На прошлом уроке вы познакомились с типами данных. Определение переменной, логическими выражениями. Приведите свои примеры задач.	

Середина урока
___ мин

Откройте тетради и запишите тему урока: «Условный оператор if».

На прошлом уроке мы научились составлять линейные программы на Python. Сегодня мы изучим конструкцию «ветвление» или «условный оператор if».

Если перевести на русский язык, конструкция условного оператора означает следующее:

Если <выполняется условие> делать: какие-то действия.

Например:

```
if a>b:  
    print(a)
```

«Если a больше b, то вывести a».

Или:

```
if x==y:  
    z=x+y  
    z=z*z
```

«Если x равен y, то z присвоить значение x+y, и возвести z в квадрат».

Отступы важны! Они – часть кода. Действия будут выполняться только в случае, если все они записаны с отступами, и причём с одинаковым количеством отступов. Стандартно в Python-сообществе принято делать 4 отступа.

Общая форма записи неполной формы условного оператора:

```
if <условие>:  
    <действие 1>  
    <действие 2>  
    и т.д.
```

Задача. Что будет напечатано в результате работы программы?

```
a=7  
b=9  
if a>b:  
    print(a)
```

(Ответ: ничего)

Неполная форма условного оператора

Русским языком:

Если <выполняется условие> делать: какие-то действия.

Пример 1:

```
if a>b:
    print(a)
```

Пример 2:

```
if x==y:
    z=x+y
    z=z*z
```

if – «если» в переводе с английского

Отступы важны! Они – часть кода. Стандартно в Python-сообществе принято делать 4 пробела.

Общая форма записи:

```
if <условие>:
    <действие 1>
    <действие 2>
    и т.д.
```

Задача. Что будет напечатано в результате работы программы?

```
a=7
b=9
if a>b:
    print(a)
```

Запись в тетрадь!

Неполная форма условного оператора

Общая форма записи:

```
if <условие>:
    <действие 1>
    <действие 2>
    и т.д.
```

Пример:

```
if a>b:
    print(a)
```

4 пробела!

Это была неполная форма условного оператора. Но у условного оператора есть и полная форма. Русским языком она звучит так:

Если <выполняется условие>: делать какие-то действия. Иначе: делать действия.

Иначе означает «если условие не выполняется».

Например:

```
if a>b:
    print(a)
```

```
else:
```

```
    print(b)
```

«Если а больше b, то вывести а, иначе вывести b.

Общая форма записи неполной формы условного оператора:

```
if <условие>:
```

<действия 1>

else:

<действия 2>

Задача. Что будет напечатано в результате работы программы?

a=8

b=5

if a<b:

print(a)

else:

print(b)

Полная форма условного оператора

Русским языком:

Если <выполняется условие> делать: какие-то действия.

Иначе: делать другие действия.

Пример:

```
if a>b:
    print(a)
else:
    print(b)
```

else – «иначе» в переводе с английского

Задача. Что будет напечатано в результате работы программы?

Общая форма записи:

```
if <условие>:
    <действия 1>
else:
    <действия 2>
```

```
a=8
b=5
if a<b:
    print(a)
else:
    print(b)
```

Запись в тетрадь!

Полная форма условного оператора

Общая форма записи:

```
if <условие>:
    <действия 1>
else:
    <действия 2>
```

Пример:

```
if a>b:
    print(a)
else:
    print(b)
```

Часто встречаются задачи с большим количеством условий и действий, когда нужно произвести при выполнении этих условий. Конструкции if-else не хватает, и на помощь приходит оператор elif. Русским языком он объясняется так:

Если <выполняется условие 1>: делать такие-то действия.

Иначе если <выполняется условие 2>: делать другие действия.

Иначе если <выполняется условие 3>: делать третьи действия.

Иначе: делать что-то ещё.

Последнее «иначе» означает «если никакие из вышеперечисленных действий выполняются». Присутствие «иначе» не обязательно. Например:

```
cost = 1500
if cost < 1000:
    print ( "Скидок нет" )
elif cost < 2000:
    print ( "Скидка 2%" )
elif cost < 5000:
    print ( "Скидка 5%" )
else:
    print ( "Скидка 10%" )
```

Что будет напечатано в результате работы программы? (Ответ: Скидка 2%)

Оператор elif

Русским языком:

Если <выполняется условие 1>: делать такие-то действия.

Иначе если <выполняется условие 2>: делать другие действия.

Иначе если <выполняется условие 3>: делать третьи действия.

Иначе: делать что-то ещё.

Пример:

```
cost = 1500
if cost < 1000:
    print ( "Скидок нет." )
elif cost < 2000:
    print ( "Скидка 2%." )
elif cost < 5000:
    print ( "Скидка 5%." )
else:
    print ( "Скидка 10%." )
```

Общая форма записи:

```
if <условие>:
    <действия 1>
elif <условие>:
    <действия 2>
elif <условие>:
    <действия 3>
...
else:
    <действия n>
```

Что будет напечатано?

Запись в тетрадь!

Оператор `elif`

Общая форма записи:

```
if <условие>:  
    <действия 1>  
elif <условие>:  
    <действия 2>  
elif <условие>:  
    <действия 3>  
...  
else:  
    <действия n>
```

Пример:

```
cost = 1500  
if cost < 1000:  
    print("Скидок нет.")  
elif cost < 2000:  
    print("Скидка 2%.")  
elif cost < 5000:  
    print("Скидка 5%.")  
else:  
    print("Скидка 10%.")
```

Знаки отношений:

> больше

< меньше

== равно

>= больше или равно

<= меньше или равно

!= не равно

Запись в тетрадь!

Знаки отношений:

> больше

< меньше

== равно

>= больше или равно

<= меньше или равно

!= не равно

Сложные условия.

Чтобы составить сложное условие используются операторы:

and - «и»

or - «или»

not - «не»

Например:

```
if a>0 and a<10 or a==100:
```

print(a)

Будет ли напечатано a, если a равно 7? А если a равно 20?

Приоритет :

- 1) отношения (<, >, <=, >=, ==, !=)
- 2) not («НЕ»)
- 3) and («И»)
- 4) or («ИЛИ»)

Сложные условия

Чтобы составить сложное условие используются операторы:

and - «И»
or - «или»
not - «не»

Пример: `if a>0 and a<10 or a==100:
print(a)`

Будет ли напечатано a, если a=7? А если a=20?

Приоритет:

- 1) отношения (<, >, <=, >=, ==, !=)
- 2) not
- 3) and
- 4) or

Запись в тетрадь!

Сложные условия

and - «И»
or - «или»
not - «не»

Пример: `if a>0 and a<10 or a==100:
print(a)`

Приоритет:

- 1) отношения (<, >, <=, >=, ==, !=)
- 2) not
- 3) and
- 4) or

Поделиться на 4 группы и решить задачи в группах

Задача «Минимум из трех чисел»

Даны три целых числа. Выведите значение наименьшего из них.

Входные данные	Правильный ответ
5 3 7	3
10 30 4	4
-5 -3 -3	-5

Задача «Ход ладьи»

Шахматная ладья ходит по горизонтали или вертикали. Даны две различные клетки шахматной доски, определите, может ли ладья попасть с первой клетки на вторую одним ходом. Программа получает на вход четыре числа от 1 до 8 каждое, задающие номер столбца и номер строки сначала для первой клетки, потом для второй клетки. Программа должна вывести YES, если из первой клетки ходом ладьи можно попасть во вторую или NO в противном случае.

Входные данные	Правильный ответ
4 4 5 5	NO
4 4 5 4	YES

Конец урока
___ мин

Рефлексия:

1. Что мы сегодня узнали?
2. Какие у вас были затруднения при выполнении задач?
3. Что у вас получилось?
4. Что у вас не получилось?
5. Как это можно исправить?

Домашнее задание

Задача №5 стр 55

Дифференциация – каким образом Вы планируете оказать больше поддержки? Какие задачи Вы планируете поставить перед более способными учащимися?

Оценивание – как Вы планируете проверить уровень усвоения материала учащимися?

Охрана здоровья и соблюдение техники безопасности

Более способные учащиеся могут выступить в роли консультантов по новой теме. Помогать слабым учащимся. Улучшать свои проекты.

Самооценивание на каждом этапе работы

Соблюдение регламента при

	на уроке Оценивание каждой из решенных задач, начисление баллов.	<i>работе</i> <i>на</i> <i>компьютере.</i> <i>Активные виды</i> <i>деятельности.</i>
--	---	---

