<table>
<tr><td colspan="3">Школа:</td></tr>
<tr><td>Дата:</td><td colspan="2">ФИО учителя:</td></tr>
<tr><td>Класс:</td><td>Участвовали:</td><td>Не участволвали:</td></tr>
<tr><td colspan="3">Тема урока: Параметры и аргументы функции.</td></tr>
<tr><td>Цели обучения, которые достигаются на данном уроке</td><td colspan="2">Познакомить с понятиями функциями и способами вызова.
Показать механизм передачи параметров</td></tr>
<tr><td>Цели урока</td><td colspan="2">Все учащиеся смогут:
• Знать конструкцию подпрограмм для создания пользовательских функций
• Уметь вызывать функции в программе
Большинство учащихся смогут:
• Работать с формальными и фактическими переменными
Некоторые учащиеся смогут:
• Составлять программы с использованием подпрограмм (пользовательские функции)</td></tr>
<tr><td>Критерии оценивания</td><td colspan="2">Понимание: умение вызывать стандартные функции
Умеет составлять программы с использованием подпрограмм</td></tr>
<tr><td>Воспитание ценностей</td><td colspan="2">Духовное развитие, уважение друг к другу, взаимопонимание</td></tr>
<tr><td>Предварительные знания</td><td colspan="2">Учащиеся работают над своим уровнем подготовленности к программированию</td></tr>
<tr><td>Межпредметные связи</td><td colspan="2">математика</td></tr>
<tr><td>Запланированные этапы урока</td><td>Запланированная деятельность на уроке</td><td>Ресурсы</td></tr>
<tr><td>Начало урока
2 мин</td><td>Организационный момент
Приветствие учащихся.
Объявление темы урока, целей обучения, совместное определение целей урока и критериев оценивания</td><td>слайд</td></tr>
<tr><td>Середина урока
3 мин

10 мин</td><td>Переход к теме
Объединение в группы.
Обсуждение с классом.
- Почему вы объединились именно так?

II. Обобщение и систематизация знаний.
Найти соответствие функций<table><tr><td>abs</td><td>возвращает число с плавающей точкой, округленное д цифр после запятой (по умолчанию).</td></tr><tr><td>round</td><td>возвращает значение x в степени y</td></tr><tr><td>pow</td><td>возвращающую абсолютное значение</td></tr><tr><td>int()</td><td>возвращает число с плавающей точкой, построенное числа или строки</td></tr><tr><td>float()</td><td>возвращает целочисленный объект, построенный из чи или строки11, или 0, если аргументы не переданы</td></tr></table>Деление класса на подгруппы
III. Изложение нового материала.
Разберемся теперь, как создавать функции в отдельном файле и вызывать их
Создадим файл myprog.py, содержащий следующий код (тело функции должно отделяться четырьмя пробелами)13: def f(x): x = 2 * x return x Запустим программу с помощью F5. Увидим, что в интерактивном режиме программа выполнилась, но ничего не вывела на экран. Правильно, ведь мы не вызвали функцию!</td><td></td></tr>
</table>

	Обновленная версия файла myprog.py будет иметь вид: def f(x): x = 2 * x return x print(f(4)) # комментарии игнорируются Python print(f(56)) Запустим программу с помощью F5 и увидим, что в интерактивном режиме отобразился результат! Теперь поговорим об области видимости переменных. Ранее мы сказали, что переменная является локальной (видна только внутри функции), если значение ей присваивается внутри функций, в ином случае – переменная глобальная, т.е. видна (к ней можно обратиться) во всей программе, в том числе и внутри функции. Рассмотрим пример. В отдельный файл с именем myprog.py поместим следующий код: a = 3 # глобальная переменная print('глобальная переменная a = ', a) y = 8 # глобальная переменная print('глобальная переменная y = ', y) def func (): print('func: глобальная переменная a = ', a) y = 5 # локальная переменная print('func: локальная переменная y = ', y) func() # вызываем функцию func() print('??? y = ', y) # отобразится глобальная переменная Обращаю внимание, что у функции print() могут быть несколько аргументов, заданных через запятую. В одинарные кавычки помещается строка14 . После выполнения программы получим следующий результат: >>> ======= RESTART: C:/Python35-32/myprog.py ========= глобальная переменная a = 3 глобальная переменная y = 8 func: глобальная переменная a = 3 func: локальная переменная y = 5 ??? y = 8 >>> Внутри функции мы смогли обратиться к глобальной переменной a и вывести ее значение на экран. Далее внутри функции создается локальная переменная y, причем ее имя совпадает с именем глобальной переменной – в этом случае при обращении к y выводится содержимое локальной переменной, а глобальная остается неизменной. Как быть, если мы хотим изменить содержимое глобальной переменной внутри функции? Ниже показан пример такого изменения с использованием ключевого слова global15: x = 50 # глобальная переменная def func(): global x # указываем, что x-глобальная переменная print('x равно', x) x = 2 # изменяем глобальную переменную print('Заменяем глобальное значение x на', x) func() print('Значение x составляет', x)	
8 мин	Работа в группе Упражнение 1 Создайте в отдельном файле функцию, переводящую градусы по шкале Цельсия в градусы по шкале Фаренгейта по формуле: TF = 9/5 * TC + 32	

5 мин	Рассмотрим несколько полезных особенностей при работе с функциями в Python.
	Имена функций в Python являются переменными, содержащими адрес объекта типа функция, поэтому этот адрес можно присвоить другой переменной и вызвать функцию с другим именем.
	def summa(x, y):
	return x + y
	f = summa
	v = f(10, 3) # вызываем функцию с другим именем
	Параметры функции могут принимать значения по умолчанию:
	def summa(x, y=2):
	return x + y
	a = summa(3) # вместо y подставляется значение по умолчанию
	b = summa(10, 40) # теперь значение второго параметра равно	

10 мин	40
	Ранее мы сказали, что имя функции – обычная переменная, поэтому можем передать ее в качестве аргумента при вызове функции:
	def summa(x, y):
	return x + y
	def func(f, a, b):
	return f(a, b)
	v = func(summa, 10, 3) # передаем summa в качестве аргумента
	Этот пример демонстрирует, как из функции func() можно вызвать функцию summa().
	Помимо этого, в момент вызова функции можно присваивать значения конкретным параметрам (использовать ключевые аргументы):
	def func(a, b=5, c=10):
	print('a равно', a, ', b равно', b, ', a c равно', c)
	func(3, 7) # a=3, b=7, c=10
	func(25, c=24) # a=25, b=5, c=24
	func(c=50, a=100) # a=100, b=5, c=50
	Ошибкой будет являться вызов функции, при котором не задан аргумент a, т.к. для него не указано значение по умолчанию.
	Работа в группах
	Упражнение 2 Напишите функцию в отдельном файле, вычисляющую среднее арифметическое трех чисел. Задайте значения по умолчанию, в момент вызова используйте ключевые аргументы.	

Конец урока **2 мин**	**Домашнее задание** Упражнение 3 Создайте в отдельном файле функции, вычисляющие площадь и периметр квадрата. **Рефлексия.** **Метод «Рефлексивные карточки»** **Было интересно** **Я понял что,** **У меня получилось** **Расскажу дома, что** Ученики анализируют деятельность на уроке, описывают затруднения, предлагают пути их преодоления.	

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**
Дифференциация в подборе заданий, в ожидаемом результате от конкретного ученика, в оказании индивидуальной поддержки учащемуся на этапе решении задач.	Взаимооценивание (по результатам эксперимента) Самооценивание (решение задач)	Соблюдение Правил техники безопасности в кабинете информатики