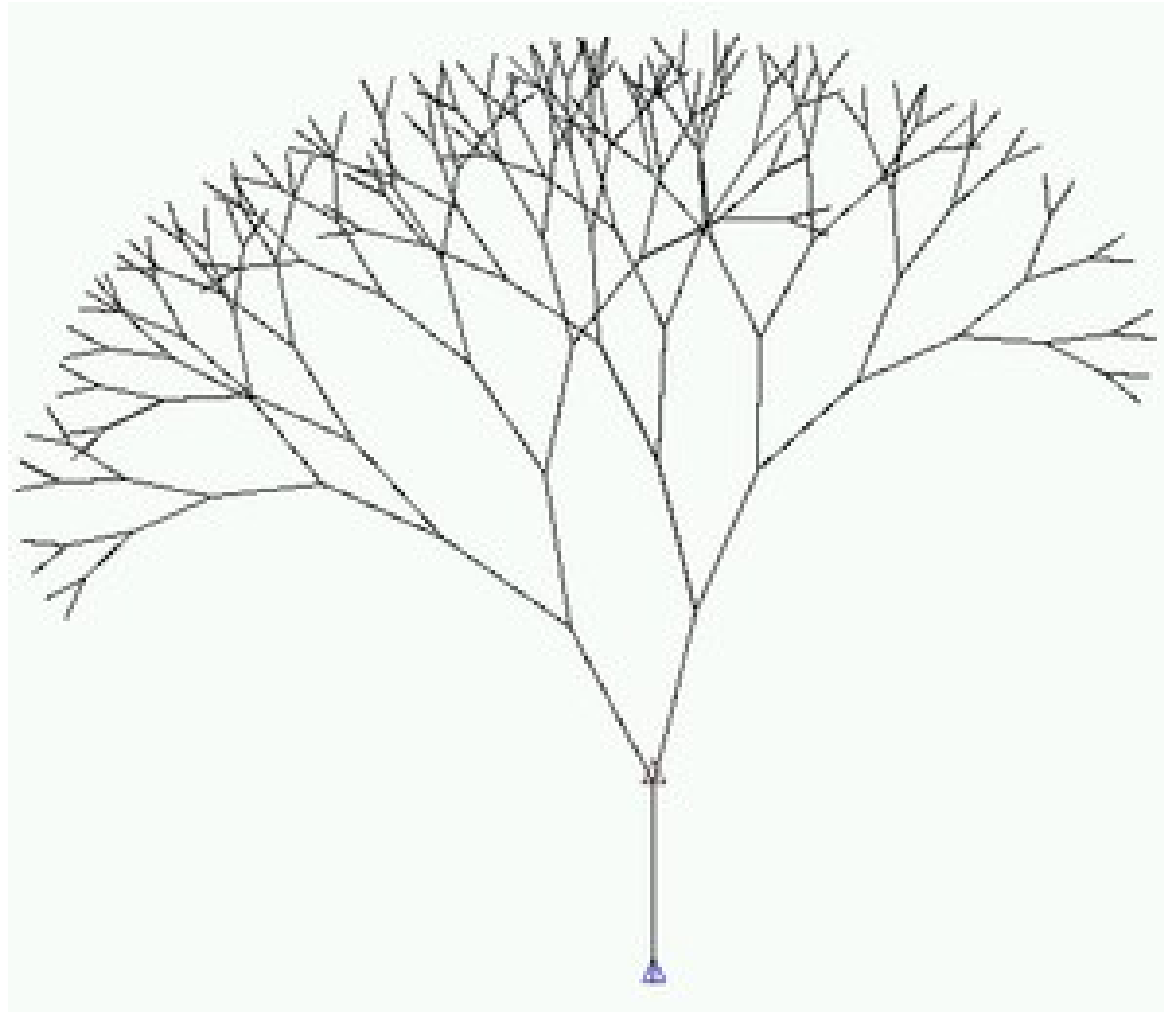


# Ветвление: основы



# Ветвление: основы

## Синтаксис:

```
if [условие ]
```

```
then
```

```
    # действия, если условие истинно
```

```
fi
```

## Условия (строки):

```
-z <строка> # строка пуста
```

```
-n <строка> # строка не пуста
```

```
<стр1> == <стр2> # строки равны
```

```
<стр1> != <стр2> # строки не равны
```

# Ветвление: основы

## Пример:

```
if [[ -z $1 ]]
then
    echo "Первый аргумент пустой!"
fi
```

```
if [[ $var1 == $var2 ]]
then
    echo "Переменные равны!"
fi
```

# Ветвление: основы

## Условия [числа (строки)]:

<число/строка> операция <число/строка>

-eq	==	#	равно
-ne	!=	#	не равно
-lt	<	#	меньше
-le		#	меньше или равно
-gt	>	#	больше
-ge		#	больше или равно

операции вида “-xx” – **только** для чисел!

операции-знаки – **только** для строк!

# Ветвление: основы

## Условия (файлы):

- e <путь> # путь существует
- f <путь> # это файл
- d <путь> # это директория
- s <путь> # размер файла больше 0
- x <путь> # файл исполняемый

## Условия (логические):

- ! # отрицание логического выражения
- && # логическое «И»
- || # логическое «ИЛИ»

# **Ветвление: альтернативы**

# Ветвление: альтернативы

## Синтаксис:

```
if [[ условие ]]  
then  
    # действия, если условие истинно  
else  
    # действия, если условие ложно  
fi
```

# Ветвление: альтернативы

## Синтаксис:

```
if [[ условие1 ]]
then
    # действия, если условие1 истинно
elif [[ условие2 ]]
then
    # действия, если условие1 ложно,
    # но условие2 истинно
else
    # действия, если оба условия ложны
fi
```



# Ветвление: альтернативы

## Синтаксис:

```
case переменная in  
знач1)  
    # действия, если переменная==знач1  
    ; ;  
знач2)  
    # действия, если переменная==знач2  
    ; ;  
*)  
    # действия, если переменная  
    # не равна ни одному из вариантов  
esac
```

# Циклы

# Циклы

## Синтаксис:

```
for переменная in список_значений
do
    # действия, каждый раз переменная
    # принимает следующее значение
    # из списка
done
```

# Циклы

## Синтаксис:

**for** переменная **in** список\_значений  
**do**

# действия, каждый раз переменная  
# принимает следующее значение  
# из списка

**done**

**break** прервать выполнение

**continue** перейти на след. значение

# Циклы

## Синтаксис:

```
while [[ условие ]]
```

```
do
```

```
    # действия, пока условие истинно
```

```
done
```

# Циклы

## Синтаксис:

```
while [[ условие ]]  
do
```

```
    # действия, пока условие истинно  
done
```

```
read переменная  
    записать введенное пользователем  
    значение в переменную
```