

```
n start()
```

```
today = Date();  
h = today.getHours();  
m = today.getMinutes();  
s = today.getSeconds();
```

```
correctTime(r);
```

```
correctTime(s);
```

```
document.getElementById("clock")  
calling the function every
```

```
t = setTimeout(start, 1000)
```

```
if (h < 10) h = "0" + h;  
if (m < 10) m = "0" + m;  
if (s < 10) s = "0" + s;  
document.getElementById("clock").innerHTML = h + ":" + m + ":" + s;  
correctTime(1)
```

# Введение в программирование на языке Julia

# Постановка задачи

Наша основная цель - подготовить рабочее пространство и инструментарий для работы с языком программирования Julia. Мы начнем с ознакомления с основами синтаксиса Julia на примерах.

# Установка Julia и Jupyter

Шаг 1: Установите Julia и Jupyter для вашей операционной системы. Это легко сделать, просто следуйте разделам 1.3.1 и 1.3.2 в документации.

```
In [1]: typeof(3), typeof(3.5), typeof(3/3.55), typeof(sqrt(3+4im)), typeof(pi)

      (Int64, Float64, Float64, Complex{Float64}, Irrational{:m})

In [2]: 1.0/0.0, 1.0/(-0.0), 0.0/0.0

      (Inf, -Inf, NaN)

In [3]: typeof(1.0/0.0), typeof(1.0/(-0.0)), typeof(0.0/0.0)

      (Float64, Float64, Float64)

In [4]: for T in [Int8, Int16, Int32, Int64, Int128, UInt8, UInt16, UInt32, UInt64, UInt128]
          println("$ (lpad(T,7)) :  [$ (typemin(T)) , $ (typemax(T)) ] ")
        end

      Int8: [-128,127]
      Int16: [-32768,32767]
      Int32: [-2147483648,2147483647]
      Int64: [-9223372036854775808,9223372036854775807]
      Int128: [-170141183460469231731687303715884105728,170141183460469231731687303715884105727]
      UInt8: [0,255]
      UInt16: [0,65535]
      UInt32: [0,4294967295]
      UInt64: [0,18446744073709551615]
      UInt128: [0,340282366920938463463374607431768211455]
```

```
In [5]: Int64(2.0), Char(2), typeof(Char(2))

      (2, '\x02', Char)

In [6]: convert{Int64, 2.0}, convert{Char,2}

      (2, '\x02')

In [7]: typeof(promote{Int8(1), Float16(4.5), Float32(4.1)})

      Tuple{Float32,Float32,Float32}
```

```
In [8]: function f(x)
          x^2
        end

      f (generic function with 1 method)

In [9]: f(4)

      16

In [10]: g(x)=x^2

      g (generic function with 1 method)

In [11]: g(8)

      64
```

# Работа с Jupyter

Шаг 2: Используйте Jupyter Lab для повторения примеров. В разделе 1.3.3 вы найдете примеры кода на Julia для различных операций, таких как поиск минимальных и максимальных значений, приведение типов, работа с функциями и матрицами.

```
In [10]: a=parse(Int64,"4534")

4534

In [11]: typeof(a)

Int64

In [12]: a=parse{Int, "4747", base=16}

18247

In [29]: a=parse{Float32, "32.15"}

32.15f0
```

```
In [25]: 97<21

false

In [26]: ~(16)

-17

In [27]: (9<11) || (15<10)

true

In [28]: !(true)

false

In [29]: tan(30)

-6.405331196646276

In [30]: exp(2)

7.38905609893065

In [31]: exp2(9)

512.0
```

```
M1=[ 60 26 62 54; 14 37 52 16; 22 10 42 91; 70 85 64 84 ]
M2=[ 82 32 52 96; 21 17 85 76; 14 57 96 79; 69 14 84 38 ]

println(M1+M2)
println(M1-M2)
println(M1*M2)
println(M1')
```

```
[142 58 114 150; 35 54 137 92; 36 67 138 170; 139 99 148 122]
[-22 -6 10 -42; -7 20 -33 -60; 8 -47 -54 12; 1 71 -20 46]
[10060 6652 15818 14686; 3757 4265 10209 8872; 8881 4542 13670 9648; 14217 8509 24065 21428]
[60 14 22 70; 26 37 10 85; 62 52 42 64; 54 16 91 84]
```

# Операции над матрицами и векторами

В этом разделе мы погрузимся в операции над матрицами и векторами. Вы узнаете, как выполнять операции, такие как сложение, вычитание, скалярное произведение, транспонирование и умножение на скаляр. Все это будет сопровождаться примерами.

```
In [10]: a=parse{Int64, "4534"}
```

```
4534
```

```
In [11]: typeof(a)
```

```
Int64
```

```
In [12]: a=parse{Int, "4747", base=16}
```

```
18247
```

```
In [29]: a=parse{Float32, "32.15"}
```

```
32.15f0
```

# Функция `parse()` в Julia

На этом слайде мы изучим и рассмотрим примеры использования функции `parse()` в Julia. Эта функция позволяет преобразовывать строки в числовые значения и может быть очень полезна при работе с данными разного типа.

```
In [13]: 6+2
8

In [14]: 4-3
1

In [15]: *(2, 5)
10

In [16]: 9/3
3.0

In [17]: 9\3
0.3333333333333333

In [18]: 4^2
16

In [19]: inv(2)
0.5

In [20]: div(100,10)
10
```

```
In [16]: 9/3
3.0

In [17]: 9\3
0.3333333333333333

In [18]: 4^2
16

In [19]: inv(2)
0.5

In [20]: div(100,10)
10

In [21]: mod(66,6)
0

In [22]: 111.0==11
false

In [23]: 2!=9
true

In [24]: 9>3
```

# Математические Операции в Julia

Давайте рассмотрим синтаксис Julia для базовых математических операций. На этом слайде вы найдете перечень операций, таких как сложение, вычитание, умножение и деление, а также примеры и объяснения их использования.

# Заключение

В заключение нашей презентации мы кратко рассмотрим пройденные темы и задания для самостоятельной работы. Мы также дадим вам рекомендации для дальнейшего изучения языка программирования Julia.