

MiniLang

Минимальный интерпретируемый язык выражений с интерактивным REPL. Поддерживает числа, строки, логические выражения, переменные и набор встроенных функций для математики и работы с текстом.

Запуск

После запуска приложения открывается REPL:

```
MiniLang REPL – выражения с math, logic, text functions.  
>
```

- Ввод осуществляется построчно.
- Каждая строка — выражение или инструкция.
- Для выхода используйте:

```
exit
```

Синтаксис

Команды

- Выражения
- Объявление переменных (`let`)
- Вызовы функций

Точка с запятой `;` допускается, но не обязательна.

Типы данных

Числа

```
10  
3.14  
-2.5
```

Строки

```
"hello"  
'world'
```

Поддерживаемые escape-последовательности: - `\n` — новая строка - `\t` — табуляция

Логические значения

```
true  
false
```

Переменные

Объявление и присваивание

```
let x = 10  
let name = "Alex"  
let ok = true
```

Использование

```
x + 5  
name + "!"
```

Арифметические операторы

Оператор	Описание
<code>+</code>	сложение
<code>-</code>	вычитание
<code>*</code>	умножение
<code>/</code>	деление
<code>^</code>	степень

Примеры:

```
2 + 3 * 4
(2 + 3) * 4
2 ^ 3
```

Операторы сравнения

Оператор	Значение
<code>==</code>	равно
<code>!=</code>	не равно
<code><</code>	меньше
<code><=</code>	меньше или равно
<code>></code>	больше
<code>>=</code>	больше или равно

Примеры:

```
5 > 3
x == 10
"abc" < "b"
```

Логические операторы

Оператор	Значение
<code>and</code>	логическое И
<code>or</code>	логическое ИЛИ
<code>not</code> или <code>!</code>	логическое НЕ

Примеры:

```
true and false
not true
x > 5 and x < 20
```

Встроенные функции

Математика

```
sin(x)
cos(x)
tan(x)
sqrt(x)
pow(x, y)
abs(x)
round(x)
max(a, b)
min(a, b)
```

Примеры:

```
sin(3.14)
pow(2, 8)
max(10, 20)
```

Строки

```
concat(a, b, c)
len(str)
substr(str, start)
substr(str, start, length)
upper(str)
lower(str)
trim(str)
```

Примеры:

```
concat("Hello", " ", "World")
len("test")
substr("abcdef", 1, 3)
upper("hello")
```

Условная функция

```
if(condition, valueIfTrue, valueIfFalse)
```

Примеры:

```
if(5 > 3, "yes", "no")
if(x > 0, x, -x)
```

Вывод в консоль

Функция `print` выводит значения и возвращает `null`:

```
print("Hello", x, 123)
```

Вывод:

```
Hello 10 123
```

Комментарии

Однострочные комментарии начинаются с `//`:

```
let x = 10 // комментарий
```

Примеры

```
let a = 5
let b = 10
print("sum =", a + b)

let s = upper("hello")
print(s)

if(a < b and b < 20, "ok", "fail")
```

Завершение

Для выхода из REPL:

```
exit
```