

Section 4: Technical Writing

Introduction

Catscript is a statically typed language which compiles on java virtual machine. the operators that catscript accepts are as follows: >,<,<=,>=,!=,+,*,/,not. the supported statements are as follows:

ForStatement,IfStatement,Functions,PrintStatement,VariableStatement,ReturnStatement. the types catscript supports are, int,bool,string,object,list<>.

Features

For loops

```
for(x in ["a","b","c","d"]){  
    print(x)  
}
```

the above is an example for loop in catscript, that will iterate through a list. in the for-loop we have our variable x that will become each value inside the list. any number of statements can be inside-of the curly braces.

If statement

```
if(x < 10){  
    print(x)  
}  
else  
{  
    print("10")  
}
```

the above is an example if statement in catscript, that will evaluate on a boolean expression. if the expression is true, the statements in the if block will be executed, otherwise the statements in the else block will be executed. the else block is not required.

Print statement

```
print("x")  
print(null)  
print(1)
```

print statement is a catscript function that prints values to the console and it only takes one argument. everything that prints in the print statement will end in a new line character.

Variable statement

```
var s = "a"  
var numb = 1
```

variable statements are used to declare variables. variables can be assigned to any data types for example you can have a variables assigned to a number also you can have a variable assigned to a string.

Return statement

```
function f(a: string): string{  
    return a  
}  
function ff(b: string){  
    if(b=="abc"){  
        return  
    }  
    print(b)  
}
```

all functions that have a non-void return type require a return statement for all control paths. void functions can optionally early returns.

Functions

```
function f(a: string): string{  
    return a  
}  
function ff(b: string){  
    print(b)  
}  
// to call a function  
f("abc")
```

all functions require names, to declare a function you must start with a function keyword followed by a functions name and ending with zero or more parameters enclosed in parenthesised. the function body must be enclosed in curley bracers and can have any number of statements. functions without a return type are assumed to be void. to declare a return type add : type before the opening curly brace to specify the return type.

Operators

Additive expressions

```
1+1 // evaluates to 2
2-1 // evaluates to 1
2+"B" // evaluates to 2B
"A"+"B" // evaluates to AB
"A"+null // evaluates to null
```

the additive expression uses the + or - operator. you can concatenate string with the + operator

Comparison expression

```
4>2 // evaluates to true
4<2 // evaluates to false
4 >= 2 // evaluates to true
4 <= 2 // evaluates to false
```

comparison expression compares to integers and returns a boolean value.

Equality expressions

```
4==4 // evaluates to true
4!=4 // evaluates to false
4 == true // evaluates to false
4 == null // evaluates to false
```

the equality expression returns boolean value, the compared values can be any data type, but any values of different data types will not be equal.

Factor expressions

```
4/2 // evaluates to 2
2*2 // evaluates to 4
```

the factor expression require both operands to be integers. they perform numerical division of multiplication

Assignment operator

```
var a = true  
a = false // valid assignment  
a= 4 // not a valid assignment
```

the assignment operator stores a variables value. only values of the variables type can be assigned to it. null be can be assigned to variables at any type. variables of object type can be assigned to any value.

Unary operators

```
-4  
not true
```

there two types of unary operators which is minus and not. minus makes numerical values negative, and not makes a boolean value its opposite.