TITANIUM LANGUAGE PROGRAMMING CHEATSHEET

- ## CONSTANTS AND VARIABLES

Constants are defined with the DEF keyword and can’t be changed later on. Variables are defined with the DECL keyword.

```
DEF name = “Matheus”

DECL age = 17
```

Titanium currently supports the following data types: STRING, BOOLEAN and NUMBER. Strings should be encapsulated between “” and boolean values can be either TRUE, YES, FALSE, or NO. Titanium also supports positive, negative, integer, float and double numbers. You can also assign NULL or UNDEF to them.

- ## COMMENTS

Titanium supports both inline and multiline comments. Single-line comments are declared by using – before the line you want to turn into a comment. A multiline comment begins and ends with $$. i.e.:

```
--This is a single-line comments
```

```
$$

This is a comment

that spaws over several lines

$$
```

- ## MATH AND LOGIC EXPRESSIONS

You can assign math expressions such as 2 + 2 or 9 % 2 to variables and constants. You can also pass them as arguments to the print function to get their result printed to the screen. You can also evaluate logic expressions such as “a” == “b” or 9 != 10. They will return either TRUE or FALSE.

```
print(9 < 20)

print(2 + 2)
```

- ## THE TERNARY OPERATOR

Titanium also supports the ternary operator. You can assign it to either variables or constants.

```
DEF x = 10

DEF y = 9

DEF result = x < y ? TRUE : FALSE
```

- ## BUILT-IN FUNCTIONS

print() - outputs something to the console. It accepts variables, constants and math and logic expressions as

its parameter.

printLine() - it prints an empty line.

get() - it asks the user for input. Its argument is the variable where the data should be stored.

clear() - it clears the console.

random() - it generates a random number. It takes three parameters: the minimum value, the maximum value and a boolean (if TRUE, then it’ll round the number to integer, otherwise it’ll return it as a double).

- ## ENDING THE EXECUTION OF THE PROGRAM

You can end the execution of a Titanium program in two ways: by using the EXIT keyword or by using RET followed by the return value, which should be a number, a math expression or a string.

- ## ARRAYS

You can declare an array in Titanium by using the ARR keyword. You should put all the data and use colons to separate the elements. You can currently store numbers and strings (???) inside Titanium arrays. Example:

```
ARR names = [“Karl”, “Doug”, “Matthew”, “Andrew”]
```

You can print all the elements in the array by using the `print()` function and passing the name of the array as its argument. To print only one of its elements, just put the index of the element you want to access inside square brackets. Example:

```
print(names[2])
```

IMPORTANT!: The first index of an array is 0, just like in most programming languages, such as C, JavaScript, and so on!

Also, before trying to access a specific index, please double-check if it exists, or else Titanium will throw an error. To get the length of an array, you can use the MAX or LEN word instead of the number of the index. Example:

```
ARR names = [“Karl”, “Doug”, “Matthew”, “Andrew”]

print(names[MAX])
```

```
ARR names = [“Karl”, “Doug”, “Matthew”, “Andrew”]

print(names[LEN])
```

- ## INCREMENT AND DECREMENT

You can increment or decrement the value of a variable by 1 by using the INC or DEC keywords respectively and the name of the variable you want to manipulate. Titanium doesn’t allow incrementing or decrementing constants.

```
INC x

DEC x
```

TRY TITANIUM OUT!!!

You can see how Titanium works by going to its playground here. You can also download its source code right here.