

DRRTY

Front-end development for back-end engineers

Manager	Dylan Bamirny
Language Guru	Richard Lopez
Tester	Rania Alshafie
System Architect	Trinity Sazo

Overview

The DRRTY programming language is an imperative and declarative Python-like scripting language used to build modular HTML components. The language aims to integrate HTML and backend logic in one to create simple web pages with basic HTML components supplied by the language. DRRTY enables programmers to build more complex HTML modules similar to the functionality of the React Javascript library. The syntax follows Python syntax closely with the addition of a few operators to distinguish between HTML code and functional code. With HTML being a generally easy language to learn and understand, the language maximizes customization of HTML components in a more functional manner.

Language Details

DRRTY's primitive data types are a mix of Python's primitive data types with the addition of lists, dictionaries, and the HTML data type as a built-in class wrapping any code that will later be compiled directly into HTML. The language is in general and not explicitly typed when declaring variables. Additional types may be added or removed as the language evolves. The control-flow consists of if-else statements, for loops, and while loops.

Data Type	Description	Example
int	Primitive integer type	x = 3

float	Primitive decimal type	y = 3.14
bool	Boolean values, True or False	off = True
str	String literal type	word = "hello"
char	Character type	letter = 'c'
list	List type stores multiple items in one variable, elements can be different types	myList = [1,2,3]
dict	Store elements in key:value pairs	myDict = {"Apple": 2, "Orange": 3, "Banana": 6}
HTML	Blocks of HTML, analogous to strings of HTML source code	<> <h1> Hello World! </h1> <\>

To define a function in DRRTY, use the keyword `def`, use parentheses to define parameters, and end the line with a colon.

`return` and `render` are both keywords where `return` functions as sending output from the function to the main program and `render` sends HTML code to the HTML client.

Sample Program

```
fav_colors = {"Dylan" : "Purple",
              "Richard" : "Sky Blue",
              "Trinity" : "Green",
              "Rania" : "Orange"}

def header(title):
    return(<><h1>{title}</h1></>)
```

```
def color_table(dict):
    row = <></>
    for name, color in dict:
        row += <>
            <tr>
                <td>{name}</td>
                <td style="background-color: {color};"></td>
            </tr>
        </>




    return(<>
        <table>
            <tr>
                <th>Name</th>
                <th>Favorite Color</th>
            </tr>
            {row}
        </table>
    </>)

def all():
    return(<>
        {header("Our favorite Colors!")}
        {color_table(fav_colors)}
    </>)

render all()
```

Example Output

Our favorite Colors!

Name	Favorite Color
Dylan	
Richard	
Trinity	
Rania	