

Art : Source Code

From bitrary

Art

First and foremost, source code formatting is so personal matter, that no software developer should ever demand others to follow his or her preferences.

Every software developer has the option to hit a key combination to reformat code to his personal taste. To keep the diff tool (<http://ss64.com/bash/diff.html>) working, source code should be recursively autoformatted prior to check-in, but it doesn't really matter, what that format is, as long as all developers of the project use it. After check-out source code can be recursively auto-formatted to each developer's specific taste.

Naming conventions are only for humans, because machines can work with any code that parser can handle. What to optimize products for is always a matter of subjective taste. Whether a measure/tool/solution saves time is also subjective, but according to Douglas Crockford (https://www.youtube.com/watch?v=_EANG8ZZbRs) one way to evaluate one's own style is, whether the style helps to prevent software flaws (bugs) and saves time by minimizing the number of software flaws.

Contents

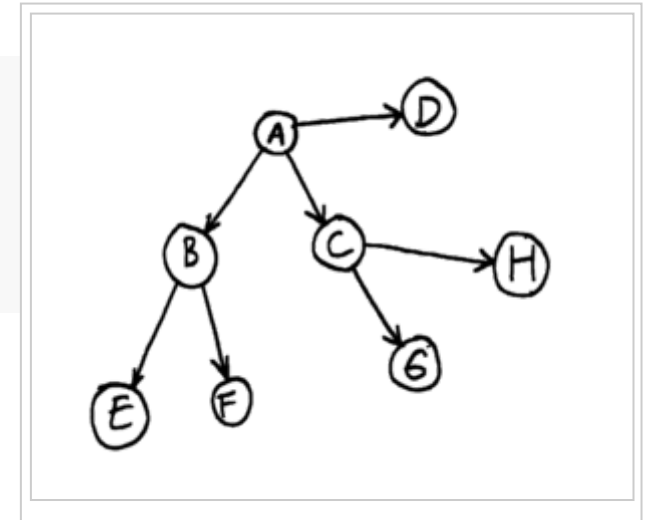
- 1 Variable Names
 - 1.1 Tree Encoding
 - 1.1.1 Type Based Prefixes (Hungarian Notation)
 - 1.2 Global Variables Versus Local Variables
- 2 Some Further Ideas
 - 2.1 Guess What
 - 2.2 Various Labeling Ideas

Variable Names

Tree Encoding

Vertices of a tree can be encoded as

```
a
a_d
a_c
a_c_h
a_c_g
a_b
a_b_e
a_b_f
```



Type Based Prefixes (Hungarian Notation)

When writing and reading code, even for functions that fit to a single screen it saves time, if the type of the variable does not have to be checked from lines that initialize the variable. One way to allow that (for humans) is to give type specific prefixes to variables.

```
i_size=42
fd_distance=33.52
ar_packages=Array.new
ht_data=Hash.new # hashtable
ob_container=MyAppSpecificClass.new # instance
s_name="This is a string"
b_do_it=true # Boolean

ix_array_index=42 # whole number, but with semantics
sb_option_is_set="t" # Boolean, but as a string that is either "t" or "f"
si_whole_number_as_a_string="993497897939883484949858211"
```

Sometimes that principle can be used for prefixing function/method names.

```
def exc_this_method_can_occasionally_throw_an_exception
  # do something here that throws at some circumstances
end # func
```

Global Variables Versus Local Variables

The syntax of some programming languages, for example, C++, JavaScript, C, does not enforce visual difference between different types (local, global, static, instance, etc.) of variables. One custom is to add "_" to the end of instance and class variables and to omit the suffix "_" from the end of local variables.

```
class_of_some_pseudoprogramming_language Awesome{
    var i_instance_variable=42; // is suffixed with "_"
    function func_music(){
        var i_local_variable=42; // lacks the suffix "_"
    }
} // end of class declaration
```

Some Further Ideas

Guess What

Example source:

```
func_nice(a,b){
  return c=a+b;
}

i_something_important=i_know_that+i_this_also_caught_my_attention+
func_nice(i_took_some_time_to_find_its_origins,i_you_guessed_it)

... // may be some other code

// Over here the question might be, did the i_something_important
// have the func_nice result added to it or not,
```

```
// specially as the func_nice call might have been  
// deleted as "unnecessary" "dead code".
```

The idea is that operators at the end of the line might be hard to notice, specially when reading code that is foreign or old.

Various Labeling Ideas

A citation from a systems programming related blog post (<https://www.bitc-lang.org/node/9>) of Jonathan S. Shapiro:

Rule: Operations that copy data should be signalled by explicit syntax.

Goal: It would be very nice to have a way to capture the whole or partial immutability of subgraphs as part of type.

Retrieved from "https://bitrary.softfl.com/index.php?title=Art:_Source_Code&oldid=1623"

-
- This page was last modified on 28 June 2015, at 16:31.
 - Content is available under Creative Commons Attribution Share Alike unless otherwise noted.