Don't be a <blank> programmer!

or: what I wish I would have known before/during/after University

SYNTAX vs. SEMANTICS

Syntax: The order in which symbols, words, or phrases are combined to form programs.

"You have a syntax error, dumbf**k.""

— *the Urban Dictionary*

This means the same thing...

```
#include <iostream>

 int main()
 {
    std::cout << "Hello, World.";
 }
```

...as this

```
function hello() {
  console.log('Hello, world!');
}

hello();
```

...and this

```
++++++++++[>+++++++>++++++++++>+++>+<<<<-]
>++.>+.+++++++..+++.>++.<<+++++++++++++++.
>.+++.------.--------.>+.>.
```

…and even this

```
Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook! Ook? Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook! Ook! Ook? Ook! Ook? Ook.
Ook! Ook. Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook! Ook? Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook?
Ook! Ook! Ook? Ook! Ook? Ook. Ook. Ook. Ook! Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook! Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook! Ook. Ook. Ook? Ook. Ook? Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook? Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook! Ook! Ook? Ook! Ook? Ook. Ook! Ook.
Ook. Ook? Ook. Ook? Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook? Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook? Ook! Ook! Ook? Ook! Ook? Ook. Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook.
Ook? Ook. Ook? Ook. Ook? Ook. Ook? Ook. Ook! Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook! Ook. Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook.
Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook!
Ook! Ook. Ook. Ook? Ook. Ook? Ook. Ook. Ook! Ook.
```

"Hi, I'm Chris, and I'm a *JavaScript* programmer."

"Hi, I'm Chris, and I'm a *Crayola* artist."

Quantitative != Qualitative

"The power to understand and predict the quantities of the world should not be restricted to those with a freakish knack for manipulating abstract symbols."

— Bret Victor

Semantics: the branch of linguistics and logic concerned with meaning.

"The purpose of internet forums is to argue over *semantics*."

— *the Urban Dictionary*

If I wrote this...

```
#include <iostream>>

 int main() {
     std:cout "Hello, World.'
}
```

I'd get this...

```
 gcc -o test test.cc
test.cc:1:20: warning: extra tokens at end of #include directive
      [-Wextra-tokens]
#include <iostream>>
                   ^
                   //
test.cc:4:11: warning: missing terminating '"' character [-Winvalid-pp-token]
 std:cout "Hello, World.'
          ^
test.cc:4:6: error: use of undeclared identifier 'cout'; did you mean
      'std::cout'?
 std:cout "Hello, World.'
     ^~~~
     std::cout
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/../include/c++/v1/iostream:50:33: note:
      'std::cout' declared here
extern _LIBCPP_FUNC_VIS ostream cout;
                                ^
test.cc:4:10: error: expected ';' after expression
 std:cout "Hello, World.'
         ^
         ;
test.cc:4:11: error: expected expression
 std:cout "Hello, World.'
          ^
test.cc:4:26: error: expected '}'
 std:cout "Hello, World.'
                         ^
test.cc:4:6: warning: expression result unused [-Wunused-value]
 std:cout "Hello, World.'
     ^~~~
3 warnings and 4 errors generated.
```

Should I learn *X*[*]?

[*] Insert name of syntactical thing (library/language/framework) you want to learn here

YES

Should I learn *X*[*] because *Y*[**]?

[*] Insert name of syntactical thing (library/language/framework) you want to learn here

[**] One of "Just so I can get a job", "Chris said so", or "All the cool developers are doing it"

NO

What *should* I learn?

Design Patterns:
"The atoms of software engineering"

— Zen Programmer Crabl

Programming Paradigms

Testing

Teamwork

"Any fool can write code that a computer can understand. Good programmers write code that humans can understand."

— Martin Fowler, God of Design Patterns

Don't let your toolset define who you are as a programmer!

Hi, I'm Chris. I'm a *programmer*.