

Forum:[Hack Forums](#)

Thread: [is this explanation of ternary operator valid?](#)

F.A.S.T - March 4, 2019, 1:58 pm

code ([Click to View](#))

PHP Code

```
<?php
#include <iostream>

bool maxEntries()
{
    int entries = 11;
    bool users = (entries > 10) ? true : false;
    return users;
}

int main(int argc, const char* argv[])
{
    if(maxEntries())
    {
        std::cout << "Entries are greater than 10." << std::endl;
    } else {
        std::cout << "Entries are less than 10." << std::endl;
    }

    return 0;
}
```

Hi, I want to know if this explanation of the ternary Operator. Using the Code above, Is valid or not. Here goes;

Code var = (condition) ? set value if condition one : set value of condition two;

A Variable with a Condition. If the condition is something than the Value of the Variable will be something. But if it's not the value will be different. Basically assign a Variable with a value based on a condition. Is this explanation valid? I need to know this.

Also, Is there any alternative of the Following python code in C++? (For this currently I am using strings)

array with different data types ([Click to View](#))

Code var = int + "string" + float + bool

Europa - March 4, 2019, 3:03 pm

I'd say "return value if condition...." or "result of ternary expression if condition..." instead of "set," but that sounds right. When you plugged in values and tested it, did you find that your explanation was correct?

Another way to explain it is like this:

Code bool users = (entries > 10) ? true : false; is bool users = entries > 10;

Disease - March 4, 2019, 5:10 pm

(Mar 4, 2019 02:58 PM)F.A.S.T Wrote: [\[CITE\]](#) code [\(Click to View\)](#)

PHP Code

```
<?php
```

```
#include <iostream>
```

```
bool maxEntries()
```

```
{
```

```
    int entries = 11;
```

```
    bool users = (entries > 10) ? true : false;
```

```
    return users;
```

```
}
```

```
int main(int argc, const char* argv[])
```

```
{
```

```
    if(maxEntries())
```

```
    {
```

```
        std::cout << "Entries are greater than 10." << std::endl;
```

```
    } else {
```

```
        std::cout << "Entries are less than 10." << std::endl;
```

```
    }
```

```
    return 0;
```

```
}
```

Hi, I want to know if this explanation of the ternary Operator. Using the Code above, Is valid or not. Here goes;

Code var = (condition) ? set value if condition one : set value of condition two;

A Variable with a Condition. If the condition is something than the Value of the Variable will be something. But if it's not the value will be different. Basically assign a Variable with a value based on a condition. Is this explanation valid? I need to know this.

Also, Is there any alternative of the Following python code in C++? (For this currently I am using strings)

array with different data types [\(Click to View\)](#)

Code var = int + "string" + float + bool

The ternary operator can be thought of as an inline conditional expression. If the conditional validates as true, the first value is returned. Otherwise, the second value will be returned.

At least that's the most concise definition I can personally come up with.

As for your second question, what are you achieving with that Python code? You can't even run that in Python... it'll result in a TypeError if you're actually trying to use the + operator on those different data types.

lockhead - March 5, 2019, 2:27 am

(Mar 4, 2019 02:58 PM)F.A.S.T Wrote: [\[CITE\]](#) Also, Is there any alternative of the Following python code in C++? (For this currently I am using strings)

Code var = int + "string" + float + bool

This will give you a `TypeError`. I think what you're trying to do is store variables of different data types in the same array or list?

Code `var = [1, "hello", 3.14, True]`

In C++ you can use `std::tuple`

<https://en.cppreference.com/w/cpp/utility/tuple>

Europa - March 5, 2019, 4:54 pm

I looked closer to your code and just want to say that explanation is not for the ternary operator, but for expression that assigns to a variable result of ternary operator. Ternary operator itself is more than that:

Code `int a = 0, b = 0;`

`condition ? a = 1 : b = 2; // based on condition assign 1 to a or 2 to b`

`( condition ? a : b ) = 100; // based on condition assign 100 to a or b`

You then use the result to assign to a variable, so your explanation is not for ternary operator but for whole expression that uses ternary:

Code `var = condition ? value1 : value2;`

You may think this is neat picking but if somebody would read your explanation he may think that assignment to a variable is part of the ternary operator itself, but that is not the case. For example:

Code `function_call( condition ? value_on_true : value_on_false );`

Also this at least confusing: "If the condition is something" it should be "if the condition is true then first otherwise (condition is false) the second".

Hopefully that makes more sense. I don't understand your python code like others said it'll just result in a runtime error.

you_can_never_hack_me - March 6, 2019, 10:14 am

Code `int func (int i, int j) {`

`return (((i != j) ? (i > j ? (--i) : (--j)) : (i-j)) != 0) ? (func (i, j)) : (0));`

`}`

0xDEAD10CC - March 7, 2019, 8:46 pm

(Mar 5, 2019 03:27 AM)lockhead Wrote: [\[CITE\]](#) This will give you a `TypeError`. I think what you're trying to do is store variables of different data types in the same array or list?

Code `var = [1, "hello", 3.14, True]`

In C++ you can use `std::tuple`

<https://en.cppreference.com/w/cpp/utility/tuple>

Only if the size of the container needs to remain the same, otherwise it wouldn't work. You'd have to use one of the C++ types that handle things by void pointers internally or use a proxy type of some sort like `boost::any`.

6577657534735 - March 7, 2019, 10:06 pm

1 means nothing

"hello" means nothing

3.14 means nothing without context

True means nothing

Put this in a struct with named fields, not a tuple or any other utility

lockhead - March 8, 2019, 2:02 am

(Mar 7, 2019 11:06 PM)Abraluna Wrote: [\[CITE\]](#) 1 means nothing
"hello" means nothing
3.14 means nothing without context
True means nothing

Put this in a struct with named fields, not a tuple or any other utility
I'm not here to give your life meaning buddy.
I gave an example of code that would run, because OPs code neither makes sense nor runs. And
yes, I did use random values to represent
the types. You got me.

But yes use a struct when you're doing this in C++, I was just talking about Python.

F.A.S.T - *March 8, 2019, 11:38 am*

That Python code I wrote was just an example of what I was trying to ask. But lets just forget it.

Thanks I lot I finally understood what a Ternary Operator is.