Php Hello World!

HelloWorld.php

```
<?php
    echo "Hello, World!"
?>
```

STDIN

Input for the program (Optional)

Output:

```
Click on RUN button to see the output
```

PHP Online Compiler

Write, Run & Share PHP code online using OneCompiler's PHP online compiler for free. It's one of the robust, feature-rich online compilers for PHP language, running on the latest version 7. Getting started with the OneCompiler's PHP compiler is simple and pretty fast. The editor shows sample boilerplate code when you choose language as `PHP` and start coding.

Taking inputs (stdin)

OneCompiler's PHP online editor supports stdin and users can give inputs to programs using the STDIN textbox under the I/O tab. Following is a sample PHP program which takes name as input and prints hello message with your name.

```
<?php
        fscanf(STDIN, "%s\n", $name);
    echo "Hello ".$name.".\n";
?>
```

About PHP

PHP(Hypertext Preprocessor) is widely used server sripting language by Rasmus Lerdorf in the year 1994.

Key features

- Free
- powerful tool for making dynamic and interactive web pages
- can integrate with almost all popular databases like MySQL, PostgreSQL, Oracle, Sybase, Informix, Microsoft SQL Server etc.
- C like Syntax and easy to learn.
- Object oriented scripting language.
- easily embeddable into HTML
- Loosely typed language.

Syntax help

Variables

In PHP, there is no need to explicitly declare variables to reserve memory space. When you assign a value to a variable, declaration happens automatically. Variables are case-sensitive in PHP.

```
$variable_name = value;
```

Loops

1. IF Family:

If, If-else, Nested-Ifs are used when you want to perform a certain set of operations based on conditional expressions.

If

```
if(conditional-expression){
//code
}
```

If-else

```
if(conditional-expression){
//code if condition is true
} else {
//code if condition is false
}
```

Nested-If-else

```
if(condition-expression1) {
    //code if above condition is true
} elseif(condition-expression2){
    //code if above condition is true
}
elseif(condition-expression3) {
    //code if above condition is true
}
...
else {
    //code if all the conditions are false
}
```

2. Switch:

Switch is used to execute one set of statement from multiple conditions.

```
switch(conditional-expression) {
case value1:
 // code if the above value is matched
 break;  // optional
case value2:
 // code if the above value is matched
 break;  // optional
...

default:
 // code to be executed when all the above cases are not matched;
}
```

3. For:

For loop is used to iterate a set of statements based on a condition.

```
for(Initialization; Condition; Increment/decrement){
  // code
}
```

For-each:

```
// you can use any of the below syntax
foreach ($array as $element-value) {
    //code
}

foreach ($array as $key => $element-value) {
    //code
}
```

4. While:

While is also used to iterate a set of statements based on a condition. Usually while is preferred when number of iterations are not known in advance.

```
while(condition) {
 // code
}
```

5. Do-While:

Do-while is also used to iterate a set of statements based on a condition. It is mostly used when you need to execute the statements atleast once.

```
do {
  // code
} while (condition);
```

Functions

Function is a sub-routine which contains set of statements. Usually functions are written when multiple calls are required to same set of statements which increases re-usuability and modularity.

How to define a Function

```
function function_name(parameters) {
  //code
}
```

How to call a Function

```
function_name (parameters)
```