Constants

PHP hằng

Một hằng số là một định danh (tên) cho một giá trị đơn giản. Các giá trị không thể thay đổi trong kịch bản.

Một tên hằng hợp lệ bắt đầu bằng chữ cái hoặc dấu gạch dưới (không có dấu $ trước tên hằng số).

**Lưu ý:** Không giống như các biến, hằng số tự động toàn cầu trên toàn bộ kịch bản.

Syntax:

define(*name*, *value*, *case-insensitive*)

example:

<?php  
define("GREETING", "Welcome to W3Schools.com!");  
echo GREETING;  
?>

Or

<?php  
define("GREETING", "Welcome to W3Schools.com!", true);  
echo greeting;  
?>

Hằng số tự động trên toàn cầu và có thể được sử dụng trên toàn bộ kịch bản.

Ví dụ dưới đây sử dụng một hằng số trong một hàm, thậm chí nếu nó được định nghĩa bên ngoài các chức năng:

<?php  
define("GREETING", "Welcome to W3Schools.com!");

function myTest() {  
    echo GREETING;  
}  
   
myTest();  
?>

cách khai báo hằng kiểu const

const teenhang = ‘giá trị’;

Sự khác biệt cơ bản giữa hai cách là constđịnh nghĩa các hằng số tại thời gian biên dịch, trong khi defineđịnh nghĩa chúng tại thời gian chạy. Điều này gây ra hầu hết các const's nhược điểm. Một số nhược điểm của constlà:

constkhông thể được sử dụng để có điều kiện xác định các hằng số. Để xác định một hằng số toàn cầu, nó phải được sử dụng trong phạm vi ngoài cùng:

if (...) {

const FOO = 'BAR'; // invalid

}

// but

if (...) {

define('FOO', 'BAR'); // valid

}

Tại sao bạn muốn làm điều đó anyways? Một ứng dụng phổ biến là để kiểm tra xem các hằng số đã được xác định:

if (!defined('FOO')) {

define('FOO', 'BAR');

}

Kể từ PHP 5.6 consthằng cũng có thể là mảng, trong khi define()không hỗ trợ mảng nào.Tuy nhiên mảng sẽ được hỗ trợ cho cả hai trường hợp trong PHP 7.

const FOO = [1, 2, 3]; // valid in PHP 5.6

define('FOO', [1, 2, 3]); // invalid in PHP 5.6, valid in PHP 7.0

về hiệu xuất const nhanh hơn gấp 2 lần so với define

Cuối cùng, lưu ý rằng constcũng có thể được sử dụng trong một lớp hoặc giao diện để định nghĩa một [lớp liên tục](http://php.net/manual/en/language.oop5.constants.php) hoặc giao diện không đổi. defineKhông thể được sử dụng cho mục đích này:

class Foo {

const BAR = 2; // valid

}

// but

class Baz {

define('QUX', 2); // invalid

}

define tôi sử dụng cho các hằng số toàn cầu.

const tôi sử dụng cho các hằng số lớp.

Bạn không thể definevào phạm vi lớp học, và với constbạn có thể. Không cần phải nói, bạn không thể sử dụng constphạm vi bên ngoài lớp học

Ngoài ra, với const, nó thực sự trở thành một thành viên của lớp, với define, nó sẽ được đẩy đến phạm vi toàn cầu.