

原创 我是哥飞 哥飞 2022-12-09 08:30:51 广东

今天的文章给大家讲讲故事。

熟悉哥飞的朋友都知道，2016年，Fenng 宣布创业之后不久，我也宣布创业，说要做一个技术搜索引擎。



Console.log()

Fenng 的创业方向是搜索，垂直搜索，可信赖的医疗信息搜索。



是的，我开始创业了

2016年10月11日 18:38 删除





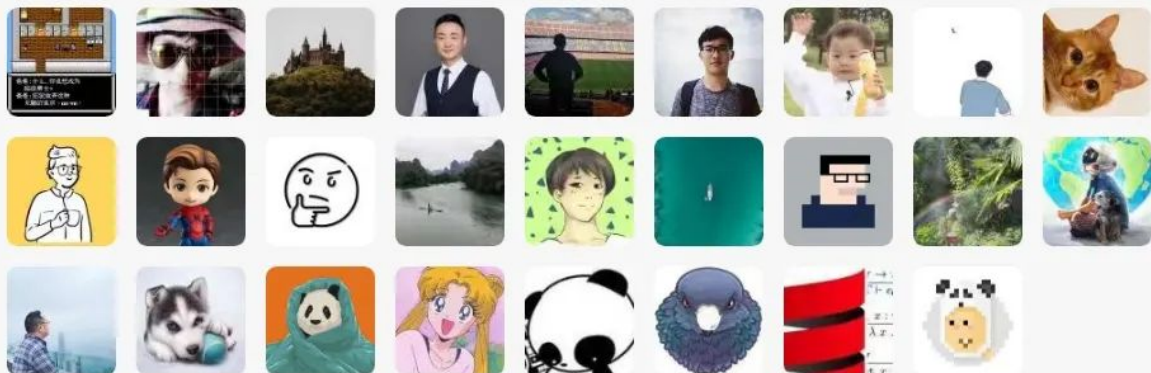
Console.log()

2017的小目标，3月1日前发布我们的面向技术开发者的垂直搜索引擎产品。如果我到时候没做到，截图给我，我给你发个100元微信红包。



是的，我也创业了，也做了一个搜索引擎

2016年12月31日 21:14 删除



Fenng

能发布吗？

2016年12月31日 21:17

Fenng 和我私聊，说本来他们也要从这个方向切入，先做一个小产品的。既然我想做这方面的产品，他们就换一换。

不久后，Readhub 上线了，果然换了方向。

可惜，我那时候忙着做H5外包，一周一个项目，一个项目收费一万多，项目就没停过，赚钱赚到手软，压根就没时间做搜索引擎，最后项目流产了。

期间，Fenng问过我一次进度，我说在忙着赚钱，一个月能赚四五万，从没见过这么多钱，我先赚钱，搜索引擎的事以后再说。

那时候为什么我想做技术搜索引擎呢？

因为我自己就有这个需求。

各大搜索引擎的中文技术搜索结果质量堪忧，技术人员深受其苦。

我就想着是否可以索引各个编程语言、开发框架、各种项目库的文档，再索引我人工挑选出来的技术博客、技术社区里的高质量帖子。

相当于人工筛选过滤优质内容提供给搜索引擎索引。

但不是简单的分词索引就完事，而是能够把信息结构化。

举个例子，当我在输入框输入“php计算数组长度函数”，不应该返回介绍函数使用的文章列表给我，而应该直接返回 [php.net](https://www.php.net/manual/en/function.count.php) 上 `count` 函数的介绍页面内容给我。

就像下面截图一样，也就是所搜即所得，不需要我在多个搜索结果里找可能符合我要求的结果，然后再点开一个新页面才能看到内容。

# count

---

(PHP 4, PHP 5, PHP 7, PHP 8)

count — 统计数组、[Countable](#) 对象中所有元素的数量

## 说明

---

```
count(Countable|array $value, int  
$mode = COUNT_NORMAL): int
```

用于统计数组中元素的数量。用于实现

用于数组时，统计数组中元素的数里，用于实现了 [Countable](#) 接口的对象时，返回 [Countable::count\(\)](#) 方法的返回值。

## 参数

---

### value

数组或者 [Countable](#) 对象。

### mode

如果可选的 **mode** 参数设为 **COUNT\_RECURSIVE**（或 1），**count()** 将递归地对数组计数。对计算多维数组的所有单元尤其有用。

**警告** **count()** 能检测递归来避免无限循环，但每次出现时会产生 **E\_WARNING** 错误（如果 array 不止一次包含了自身）并返回大于预期的统计数字。

# 返回值

---

返回 **value** 中的元素的数量。在 PHP 8.0.0 之前，如果参数既不是数组也不是实现了 Countable 接口的对象，将返回 1。当 **value** 为 **null** 时返回 0。

而如果我在输入框里输入的是“php count 函数参数”，那么我期望看到的是下面这样的。

# 参数

---

## value

数组或者 [Countable](#) 对象。

## mode

如果可选的 **mode** 参数设为 **COUNT\_RECURSIVE**（或 1），**count()** 将递归地对数组计数。对计算多维数组的所有单元尤其有用。

**警告** **count()** 能检测递归来避免无限循环，但每次出现时会产生 **E\_WARNING** 错误（如果 array 不止一次包含了自身）并返回大于预期的统计数字。

而如果我在输入框里输入的是“php count 使用示例”，那么我期望看到的是下面这样的。



# 范例

---

## 示例 #1 count() 例子

```
<?php
$a[0] = 1;
$a[1] = 3;
$a[2] = 5;
var_dump(count($a));

$b[0]  = 7;
$b[5]  = 9;
$b[10] = 11;
var_dump(count($b));
?>
```

以上例程会输出：

```
int(3)
int(3)
```

也就是说，我想做的搜索引擎能够读懂我在问什么，也能够根据我问的返回符合我要求的结果。

而这个结果哪里来？

只能是来自于从各种文档、文章里截取的结构化内容。

但是，注意，我要说但是，这依然还是属于传统搜索引擎，返回的结果来自于对已有内容的裁剪，而无法无中生有创造内容。

我们再把上面三个问题交给 ChatGPT，看看它会怎么回答。



## php计算数组长度函数



php中的数组长度函数是`count()`。例如，如果你有一个名为`$myArray`的数组，你可以用下面的代码来计算它的长度：

 Copy code

```
$arrayLength =  
count($myArray);
```

注意，这个函数不仅可以用于计算数组的长度，还可以用于计算对象的属性个数。





## php count 函数参数



count()函数有两个参数：

\$array：要计算长度的数组。

\$mode：可选参数，表示计算方式。有三个可选值：

- COUNT\_NORMAL：默认值，计算所有元素。
- COUNT\_RECURSIVE：递归计算，如果数组中的某个元素还是一个数组，则会递归计算。
- COUNT\_PHP\_5\_3：在php5.3版本中，使用此选项可以计算元素值为NULL的元素。

例如，如果要计算数组\$myArray中的元素个数，可以使用下面的代码：

 Copy code

```
$arrayLength =  
count($myArray);
```



## php count 使用示例



php中count()函数的使用方法非常简单，下面是一个简单的示例：

 Copy code

```
// 定义一个数组
$myArray = array(1, 2, 3, 4,
5);

// 计算数组的长度
$arrayLength =
count($myArray);

// 输出数组的长度
echo $arrayLength; // 输出：5
```

上面的代码定义了一个名为\$myArray的数组，然后使用count()函数来计算数组的长度，最后使用echo语句将数组的长度输出到控制台。

---

从 ChatGPT 的回答可以看到，它通过理解内容，理解我问的问题，然后用自己的话来回答了我的问题，它无中生有创造了新的内容。

无中生有，对，无中生有，这是 ChatGPT 最强大的地方。

所以，你能知道我这几天为什么这么兴奋吗？

因为 ChatGPT 实现了我曾经想要做的产品，并且做得更好，好一万倍的那种好。

ChatGPT 给人全新的体验，曾经沧海难为水，除却巫山不是云。

这就是我的感觉，用了 ChatGPT 后，再也不想有传统搜索引擎那种只能根据关键字索引找到内容的体验了。

我期待，期待着 ChatGPT 能够开放提供给用户喂语料，这样我就可以把更多的文档喂给他，进而帮助我们更好的学习、工作、生活。

[#ChatGPT](#)