برنامه نویسی مبتنی بر وب

استاد: برنجی

علیرضا سلطانی نشان

ترم سوم

1399.07.15

فهرست مطالب

تفاوت بین Programming Language و Script Language

مهم ترین تفاوت بین زبان برنامه نویسی با زبان اسکریپت نویسی، به نوع اجرای[1] آنها بر می‌گردد. در Programming Language یک مترجم[2] وجود دارد تا برنامه نوشته شده شما را از زبان سطح بالای[3] برنامه نویسی به زبان سطح پایین ماشین[4]، تبدیل کند. در زبان اسکریپت نویسی، دیگر خبری از یک مترجم نیست، بلکه یک مفسری[5] در برنامه حاکم است تا کدهای شما را به صورت خط به خط اجرا کند و در این سیستم دیگر خبری از مترجمی نیست که کاملا یک برنامه را به زبان سطح پایین ترجمه کند و سپس خروجی را به توسعه دهنده نمایش دهد.

تفاوت های اساسی PL با SL

دسته بندی‌ها[6]

تفسیر

زبان های برنامه نویسی، با یک طراحی جمع و جور طراحی شده اند در حالی که نیازی به تفسیر کننده کد توسط برنامه یا زبان دیگری ندارند، در حالی که در زبان های اسکریپتی، با یک زبان نوشته شده اند و توسط دیگر برنامه ها تفسیر می‌شوند. برای مثال اگر ما بخواهیم JavaScript را به صورت Native در کنار HTML بنویسیم، برنامه ای که این کد های جی اس را اجرا می‌کند در واقع همان مرورگر است، یا مثلا در Node.js برای اجرای بلادرنگ [9]برنامه نیاز به یک برنامه‌ی دیگری بنام nodemon داریم که بتواند کدهای نوشته شده را تفسیر کند تا مرورگر در مرحله بعدی توسط لایه نمایش در معماری MVC، قسمت سمت کاربر را نمایش دهد.

توسعه[10]

[1] Execute
[2] Compiler
[3] High Level
[4] Low Level (Machine Level)
[5] Interpreter
[6] Categories
[9] RealTime
[10] Development

نوشتن کد با استفاده یک زبان برنامه نویسی مانند سویفت یا سی، نسبتا سخت می باشد چون که برای مثال برای نوشتن یک تابع، نیاز به کد خط‌های زیادی است. در حالی که در یک زبان اسکریپتی مانند پایتون یا جی اس یا PHP، در زمان و تعداد کدهای کمتری خیلی راحت میتوان یک تابع یا کلاس را نوشت.

سرعت

سرعت برنامه های نوشته شده توسط زبان های برنامه نویسی نسبت به زبان های اسکریپتی، بیشتر است، زیرا زبان های برنامه نویسی یک بار کد را ترجمه می‌کنند اگر مشکلی یا خطایی در فرآیند اجرای برنامه وجود داشته باشد، برنامه همان اولین زمان متوقف می‌شوند، و خطا و هشدار را اعلام میکنند، در غیر این صورت اگر هیچ مشکلی نباشد کل برنامه را به صورت تمام و کمال اجرا خواهند کرد. اما در زبان های تفسیر کننده چون که برنامه به صورت خط به خط اجرا می‌شوند، اگر در یکی از خط‌ها مشکلی باشد کل برنامه متوقف می‌شود و نتیجه ناقصی را خواهیم داشت.

منظور از شئ‌گرایی[11] چیست؟

برنامه نویسی شئ‌گرا یکی از روش های برنامه نویسی است که بر پایه یکسری از مفاهیم کلاس ها و شئ ها است. شئ‌گرایی به عنوان ساختار یک برنامه نرم‌افزاری مورد استفاده قرار میگیرد، که میتوان از آنها به صورت مجدد بارها و بارها استفاده کرد، یعنی ما آنها را به صورت یک ساختار یا چهارچوب کلی ایجاد می‌کنیم و می‌توانیم نمونه های [12]زیاد و مختلفی از آن ها داشته باشیم. به خاطر اینکه شئ‌گرایی یک شیوه برنامه نویسی است ما در این زمینه زبان های زیادی مانند، سی ها، پایتون، جی اس و غیره را داریم.

[11] Object Oriented Programming

[12] Instances

یک برنامه نویس اطلاعات و رفتار هایی که می‌توانند روی این اطلاعات تاثیر داشته باشند، در نرم‌افزار خود از الگویی به نام Class استفاده می‌کند.

در نوشتن این گزارش برای معرفی شئ ها از زبان JS استفاده کردم:

```
} const users = {
    name: 'Alireza',
    family: 'Soltani',
    age:  getAge = (userBirthday) => {
        const realAge = Date.now() - userBirthday
        return realAge
    },
    field: "JS programmer"

}// This is a very simple object in JS
```

یک کلاس ساده برای دریافت نام و نام خانوادگی و سن او:

```
class users {
  constructor(name, family, birthday) {
    this.name = name
    this.family = family
    this.birthday = birthday
  }

  calculateUsers() {
    const years = Date.now() - this.birthday
      const name = this.name
      const family = this.family
      return {
        name,
        family,
        years
      }
    }
  }

const user1 = new users('Alireza', 'Soltani', 2000)
const alirezaSoltaniNeshan = user1.calculateUsers()

console.log(alirezaSoltaniNeshan)

```

فریم‌ورک

فریم‌ورک یک کتابخانه‌ای برای استفاده در اهداف خاص و مشخص است که توسط یک زبان برنامه‌نویسی معرفی شده. تا بتوانیم با استفاده از آن سرعت بیشتری برای تولید محصول و نتیجه خود داشته باشیم. برای مثال در زبان جی‌اس فریم‌ورک های زیادی در استفاده های گوناگونی وجود دارد. بر فرض مثال ما میخواهیم برنامه نویسی سمت کاربر را با یک فریم‌ورک انجام بدیم که بسیار واکنش پذیر و پویا باشد پس از فریم‌ورک وب React یا Angular استفاده می‌کنیم. یا برای برنامه نویسی موبایل می‌توانیم از زبان جی‌اس با استفاده از فریم‌ورک React Native و در زبان پایتون با استفاده از Kivy (کیوی) استفاده کنیم. یا اینکه اگر بخواهیم برنامه نویسی سمت سرور را داشته باشیم از پایتون فریم‌ورک Django یا پلتفرم Node.js استفاده می‌کنیم.

برخی از مزایای استفاده از فریم‌ورک ها:

- کاهش زمان کد زدن
- افزایش بهره‌وری و صرفه جویی در تعداد خط ها
- بسیار مناسب برای کارهای تیمی
- استفاده از مدل طراحی MVC

Editor و IDE، تفاوت در چیست؟

بطور کلی در یک IDE هنگام نصب و راه اندازی آن، یک نرم افزار به صورت یکپارچه با یکسری از تکنولوژی‌ها و فریم‌ورک ها نصب خواهند شد که ممکن است در ابتدای نصب نیازمند به اینترنت باشید که تمامی پکیج ها و برنامه های مورد نیاز دانلود شوند، و بعد از راه اندازی یک IDE، دیگر نیاز به هر بار نصب راه‌اندازی یک پلتفرم یا اولیه سازی در هر بار نوشتن یک پروژه جدید نیست. برای مثال وقتی شما برنامه Android Studio را نصب میکنید، بعد از راه اندازی آن و آماده کردن Gradle دیگر نیاز به نصب دوباره پکیج ها ندارید. در حقیقت یک IDE محیطی یکپارچه را فراهم میکند که به توسعه پلتفرم خاص خود بپردازید.

یک ویرایشگر یا (Editor) حکم یک جعبه ابزار همه فن حریف را دارد که برای هر کاری مورد استفاده قرار گیرد،(کاملا یک ویرایش کنند حالا هر چیزی) برای مثال، شما می‌توانید برنامه‌ای با هر زبانی بنویسید، به شرط آنکه تمام ملزومات و کتابخانه های آن نصب باشد، به عنوان پیشفرض VS

code، برای نوشتن یک برنامه فلاتر نیازمند نصب تمامی پکج ها در هر پروژه خود دارید، یا مثلا در هنگام برنامه نویسی سرور ساید در زبان جی‌اس ابتدا نیاز به بسته های node_moduals در هر پروژه دارید، که بایستی به وسیله پکیج منیجر npm کتابخانه ها و فریم‌ورک های موردنظر را نصب کنید. این نصب و به اصطلاح init کردن در هنگام استفاده از ویرایش به صورت یکپارچه برای همیشگی نیست بلکه هربار پروژه زدن نیاز به دوباره کاری داریم، اما در IDE مانند Xcode هنگام پروژه زدن یا init کردن برنامه خود همه وابستگی ها، همان ابتدای نصب مشخص شده اند و نیازی به دوباره کاری نیست.

انوع روش های برنامه‌نویسی

روش های برنامه نویسی، یعنی چه راه هایی برای نوشتن یک برنامه وجود دارد.

روش های برنامه نویسی هم به تعداد هستند: که در اینجا 4 نمونه از آشناترین آنها را برسی خواهم کرد.

برنامه‌نویسی ضروری[13]

کنترل جریان در برنامه‌نویسی ضروری به صورت صریح است. یعنی دستورات نحوه محاسبه را قدم به قدم نشان می‌دهند.

```
1.      result = []
2.      i = 0
3.  start:
4.      numPeople = length(people)
5.      if i >= numPeople goto finished
6.      p = people[i]
7.      nameLength = length(p.name)
8.      if nameLength <= 5 goto nextOne
9.      upperName = toUpper(p.name)
10.       addToList(result, upperName)
11.   nextOne:
12.       i = i + 1
13.       goto start
14.   finished:
15.       return sort(result)
```

[13] Imperative

برنامه‌نویسی ساختارمند[14]

برنامه‌نوسی ساختارمند هم از هم‌نوع برنامه‌نویسی ضروری میباشد با این تفاوت که کنترل جریان به صورت حلقه های تو در تو مشخص شده. مثال زیر یک نوع خیلی واضحی از برنامه ساختارمند است، در آن مشخص شده که از خانه اول آریه ای بنام people که در آن object هایی قرار دارد، افرادی که تعداد کلمه نام آنها بیشتر از 5 باشد را در لیست result پوش میکند.

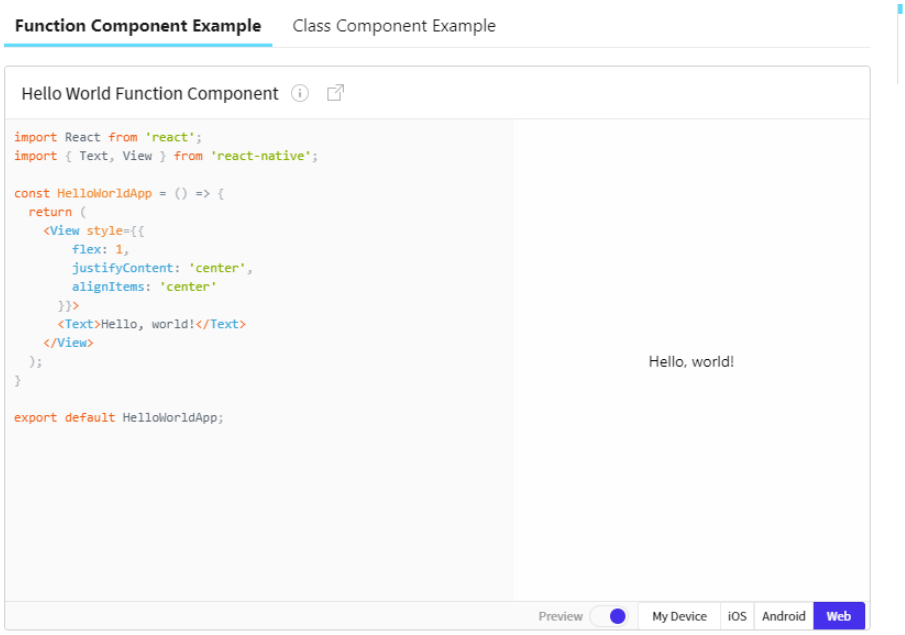
```
1. result = [];
2. for i = 0; i < length(people); i++ {
3.     p = people[i];
4.     if length(p.name)) > 5 {
5.         addToList(result, toUpper(p.name));
6.     }
7. }
8. return sort(result);
```

برنامه‌نویسی تابعی (عملکردی)[15]

در برنامه‌نویسی تابعی تمام علمیات درون یک تابع نوشته می‌شود، در این یک تابع ممکن است تعداد کم یا زیادی از توابع برای انجام یکسری Taskها وجود داشته باشد. (یک مثال) در زمانی که شروع به یادگیری برنامه نویسی جی اس کردم و زمانی که خواستم بروی فریم‌ورک ریکت نیتیو مشغول شوم، دو راه برای نوشتن برنامه های React Native با JS وجود داشت، یک برنامه نویسی extended

[14] Structed
[15] Functional Programming

Class (حالی نه مثل polymorphism) و دیگری برنامه‌نویسی به صورت Functional. تا قبل از سال 2018 برنامه نویسی با فریم‌ورک React Native به صورت تابعی، در حقیقت برای نمایش محتوای ثابت بود[16] و برای پویایی و استفاده از کلاس ها و کانکشن ها دیگر، از extended class استفاده میشد، از سال 2018 به بعد دیگر فرقی نمیکند که شما به صورت تابعی برنامه نویسی میکنید یا کلاسی، در هر دو روش می‌توانید به یک نتیجه برسید.



شکل 1اکسپو کامپایلر، روش تابعی یا کلاسی

[16] Static Contents

برنامه‌نویسی اظهاری[17]

نیازی به توضیح اضافی نیست، در حقیقت این نوع از برنامه‌نویسی را همه توسعه دهندگان با آن آشنا هستند، زبانی که به صورت واضح با استفاده از یکسری پرس و پاسخ هایی را انجام میدهیم. مانند زبان SQL.

```
1. select                                                    upper(name)
   from people where length(name) > 5 order by name
```

تاریخچه PHP

PHP یک زبان اسکریپت‌نویسی مخصوص توسعه وب می‌باشد. در حقیقت توسط برنامه نویس دانمارک-کانادایی به نام Rasmus Lerdorf در سال 1994، ساخته شده است. PHP درواقع بر پایه Personal Home Page است، اما آن امروزه بر پایه Hypertext Preprocessor است. کدهای PHP معمولا توسط یک سرویس دهنده وب مانند (درحالت لوکال) Wamp، توسط مفسر PHP پردازش می شوند و نتیجه توسط مرورگر نمایش داده میشود. در وب سرویس، نتیجه تفسیر و اجرای کد PHP که ممکن است هر نوع داده ای باشد، مانند داده هایی از قبیل HTML یا حتی Binary Image Data، که می‌خواهند توسط پروتکل HTTP به سمت کاربر پاسخ داده شوند. مانند Web Template های مختلف مانند Handlebar, EJS و غیره، PHP می‌تواند لا به لای ساختار HTML نوشته شود. PHP میتواند در بسیاری از وظایف خارج از فضای وب هم مورد استفاده برنامه نویسان قرار گیرد، از قبیل برنامه های گرافیکی مستقل[18]، یا کنترل ربات ها یا برنده های کنترلی. و همچنین مانند پلتفرم نود جی اس می‌تواند به صورت خط فرمان[19] هم مورد استفاده قرار گیرد.

استاندارد مفسر زبان PHP بر پایه و اساس Zend Engine، برنامه رایگان منتشر شده زیر نظر لایسنس PHP است. که درواقع Zend Engine یک برنامه متن باز به عنوان مفسر زبان اسکریپتی PHP است. PHP به طور گسترده و با محدوده بزرگی می‌تواند در بسیاری از وب سرویس ها و تقریبا در هر سیستم عامل و پلتفرمی به طور آزاد مورد استفاده قرار گیرد.

[17] Declarative Programming
[18] Standalone Graphical Applications
[19] Command Line Interface

نسخه های PHP

نسخه 4.0

الان PHP در نسخه 7 خود بسر میبرد. PHP از نسخه 4 شروع به توسعه شد، در 22 می سال 2000 تقریبا 18 ماه پس از اعلام رسمی در مورد ازسر گیری فعالیت‌های برنامه‌نویسی برای نسخه جدید، PHP 4.0 به بیرون منتشر شد. بسیاری از مردم، نسخه 4 را جنجالی‌ترین نسخه از این زبان می‌دانستند و در بسیاری از فروم ها صحبت هایی در مورد آن میشد. بعد از چند ماه، سایت netcraf گزارشی بیرون داد که تخمین می‌زد بیشتر از 3.6 میلیون دامنه، PHP را روی دامنه های خود نصب کرده اند.

نسخه 5.0

در نسخه پنجم PHP بسیاری از توابع اضافه شدند، توابعی مانند، ()destroy و سازنده ها، تکثیر اشیاء، class abstraction، حوزه متغیرها، رابطها و ارتقای طریقه مدیریت کردن اشیا.

نسخه 6.0

در این نسخه از زبان PHP ، از کد های Unicode پشتیبانی شد، امنیت پیشرفت بسیار زیادی داشته،

نسخه 7.0

بعد از نشخه ششم این زبان، ویژگی هایی زیادی معرفی شدند که در اینجا به چندتا از آنها اشاره می‌کنم:

مدیریت بهتر خط ها، موتور Zend بهبود پیدا کرد، از نسخه 64 بیتی پشتیبانی نمود، انواع کلاس های بدون نام، کارایی و سرعت بیشتر نسبت به قبل، ایجاد عملگرهای جدید (مخالف <>)، مرتب سازی متغیر ها و تعریف کردن مقادیر برای هرکدام، قابلیت مدیریت عیب Single Thread بودن با استفاده از Promiseها و غیره.

انوع روش های کد زنی پی اچ پی:

پایه ای

شئ گرا

فریم‌ورک MVC

نوشتن کد در زبان PHP

برای نوشتن برای های PHP می‌توانیم در لابه‌لای ساختار HTML مورد استفاده قرار گیرند، برای نوشتن یک برنامه PHP بایستی در وب سرویس شما این اتفاق بیوفتد چرا که این زبان یک زبان اسکریپتی سمت سرور می‌باشد باید با فرمت PHP در وب سرویس شما اجرا شود، مهم نیست از چه وب سرویسی استفاده می‌کنید، با کمی آگاهی فقط کافی است در آن فایلی با پسوند مربوط نوشته و در داخل آن به توسعه برنامه PHP خود بپردازید.

برای نوشتن هر کدام از کدهای PHP لازم است درون <?php?> نوشته شود تا کدهای HTML شما با PHP تمایز داشته باشند.

دستور echo

برای نمایش خروجی و یک پیام می‌توان از دستور echo استفاده کرد.

```
1. <?php
2. echo “Hello, PHP”;
3. ?>
```

دستور Print

دستور Print هم مانند دستور echo می‌باشد

```
1. <?php
2. print(“Hello, PHP”);
3. ?>
```

تفاوت echo با print

اما سرعت اجرا شدن در echo بیشتر از print است. در echo شما می‌توانید چند تا پارامتر را به عنوان نمایش وارد کنید اما در Print همچین امکانی وجود ندارد و در هنگام استفاده از echo ما هیچ مقداری به عنوان مقدار برگشتی نخواهیم داشت، اما در Print مقدار عدد 1 برگشت داده می‌شود.

متغیرها

برای معرفی متغیر به نحو زیر عمل می‌کنیم:

```
1. <?php
2. $name = "Alireza";
3. $age = 19;
4. echo $name, $age;
5. ?>
```

متغیر ها قسمتی هایی از یک حافظه هستند که با توجه به نوع مشخصی یک قسمتی را اشغال می‌کنند و به صورت موقتی تا زمان اجرای برنامه در دسترس می‌باشند.

انوع مخلفی که حتما میدانیم وجود دارد:

رشته ها، اعداد (اعشاری و صحیح)، آرایه ها و آبجکت ها فانکشن ها و غیره.

ثابت ها

زمانی که از ثابت ها استفاده میکنیم دیگر نباید قصد تغییر مقدارشان را داشته باشیم:

```
1. <?php
2. define("number", 58 );
3. echo "The result number is".number;
4. ?>
```

تفاوت Single quote و Double quote

در Single quote ما اگر متغیری را بنویسیم، فقط آنرا نمایش میدهند یعنی:

```
1. echo '$a' //$a
```

اما وقتی از double quote استفاده میکنیم مانند آن است که داریم اسم متغیر را صدا میکنیم که برای فقط مقدارش را نمایش دهد.

```
1. echo "This is a simple number in double quote: $a";//This is
   a simple number in double quote: 15
```

برای اینکه بخواهیم از نوع متغیری باخبر شویم از تابع gettype(your var) استفاده خوهیم کرد

Array آرایه‌ها:

با استفاده از آرایه ها میتوانیم مجموعه از متغیر ها با نوع و اندازه مشخصی را به صورت ذنجیره ای در حافظه به صورت موقت ذخیره کنیم، با استفاده از آرایه داده های ما چینش و قرار گیری مرتبی میتواند داشته باشد و همچنین می‌تواند به صورت چند بعدی نیز مورد استفاده قررا بگیرد.

نحوه معرفی یک آرایه در PHP:

```
1. $arr = array(1, 2, 3, 4, 5);
2. $arr[2]; // 3
3. count($arr); //return length of Array
```

تفاوت Get و Post:

برای ارسال اطلاعات از سمت کاربر به سمت سرور باید توجه داشته باشیم که میتوانیم این کار را به دو روش (متد) انجام دهیم. یک روش Get است که بهتر است بگویم برای نمایش اطلاعات می‌توانیم به راحتی از آن استفاده کنیم و برای ارسال اطلاعات از سمت کاربر به سمت سرور و درج آن در یک دیتابیس بایستی از متد Post استفاده کنیم تا ارسال اطلاعات با امنیت بیشتری صورت گیرد، اما برای توضیح موضوع امنیت باید بگویم که در متد get وقتی ارسال اطلاعات خود را به سمت سرور انجام میدهیم، تمام ریکوئست های کاربر در url bar نمایش خواهد یافت و با توجه به محدود مقدار در این آدرس بار اگر ریکوئست های کاربر زیاد باشد باعث رخ دادن مشکلاتی دیگر می‌شود اما در روش post اطلاعات در بستری امن تر ارسال میشود بدون آن که در گوشه ای نمایان شوند.

25 تگ جدید در HTML 5:

formatting:

abbr: Represent abbreviation

address: This is a sematic tag and important for SEO of your page.

b: Make your text Bold. Semantic tag.

cite: Defines the title of a work or reference of work.

code: Make your piece of computer code, (can use some syntax highlighter to have more readability)

del: Convert your minimal text with a line on over like this, ~~I'm deleted~~.

i: Define a part of text in an alternate voice or mood.

kbd: Define keyboard input area for your HTML page.

pre: New tag, if you text any things to your paragraph all things will be formatted as you write in between pre tags.

mark: make section of your text highlight.

blockquote: make your selected section quoted.

dfn: specific a term that is going to be defined within the content.

em: a semantic tag for SEO for defining a section of your text is emphasize.

ruby: define a ruby annotation

sub: subscripted text

sup: superscripted text

var: define a variable.

wbr: create a line-break on your page between some text and paragraph.

Forms and Inputs:

select: defines a drop-down list.

legend: defines a caption for a fieldset.

datalist: specifics a list of pre-defined options for input controls.

Frame:

iframe: defines an inline frame

links:

link: only defines the relationship between a document and an external sources (used for stylesheets or some scripts)

lists:

ul, li, ol(order list)

dl: defines a description list

dt: defines a term/name in a description list

dd: define a description of a term/name in a description list.

تابع intval()

این تابع می‌تواند عددی ورود را فقط قسمت integer یا قسمت صحیح را نمایش دهد.

```
1. echo(intVal(52.85)); \\ return 52
```

توابعfunction:

زمانی از توابع استفاده میکنیم که بخواهیم کد تمیز و خوانایی داشته باشیم که قابلیت ماژولاریتی داشته باشد این ویژگی باعث میشود که از تکرار کد های اضافی جلوگیری کنیم و همچنین در دیباگ کردن برنامه خود مشکلی نداشته باشیم و اصطلاحا به صورت بخش های مشخص به بررسی کد خود بپردازیم.

```
1. <?php
2. $std_name = "Alireza";
3. say_hello($std_name);
4. function say_hello($name){
5. echo "Hello $name, welcome" ;
6. }
7.
```

Variables متغیر ها

متغیر های محلی:

متغیرهایی هستند که به صورت داخلی در داخل توابع، حلقه ها، یا هر قسمتی از کدخط های شما که در داخل بلاک دیگری است، مورد استفاده قرار میگیرند.

متغیر های سراسری:

متغیر هایی هستند که در هر جایی مورد استفاده قرار میگیرند، معمولا در ابتدای کد خط یکسری متغیر هایی وجود دارند که بعد طی فرایند هایی مقدایری به انها وارد میشود یا تغییراتی رخ میدهد، اینگونه متغیرها انحصاری به ابتدای خط بودن ندارند، شما میتوانید در هر قسمتی که میخواهید متغیر خود را تعریف کنید و در براکت های دیگری که بعد از آن وجود دارند استفاده کنید بطوری که در کد خط هایی که در بالا قرار دارند این متغیر قابل استفاده نخواهد بود. معمولا شما میتوانید یک متغیر را هر جایی که دلتان میخواهد بنویسید + نوشتن عبارت global برای اینکه در کل برنامه این متغیر سراسری با دیگر متغیر ها متمایز شود.

سوپر گلوبال یا ابر سراسری:

یکسری متغیر هایی هستند که در کل برنامه میتوانند در زبان php مورد استفاده قرار بگیرند، مانند post یا get یا Session یا cookies و غیره که همه هر قسمتی که ما میخواهیم میتوانیم از آنها استفاده کنیم.

settype()

با استفاده از این تابع میتوانی نوع متغیر را تغییر دهیم.

```
1. settype($var_name, "your type will be here");
```

casting

روشی دیگر برای تغییر نوع متغیر میباشد که به آن کست کردن میگویند.

```
1. $var = 12;
2. $var = string($var);
```

strpos

این تابع زمانی کاربرد دارد که بخواهیم جایگاه اولین کلمه که مورد نظر ما هست را پیدا کنیم.

```
1. strpos($var, your signal);
```

```
1.  $name = "Alireza Soltani Neshan Alireza";
2. strpos($name, "Alireza")
```

strtolower()

تابعی که برای رشته ها مورد استفاده قرار میگرد و همان طور که از اسمش معلوم است تمام کلمات را به کوچک تبدیل خواهد کرد.

trim()

این تابع برای از بین بردن فضای خالی بین کلمات مورد استفاده قرار میگرد.

trim in get and post

index

```
1. <form action="submission.php" method="POST">
2.     <input type="text" name="username">
3.     <input type="password" name="password" id="pass">
4.     <input type="submit" value="Go to welcome">
5. </form>
```

submission page

```
1. <?php
2. include "./controller/ctr.php";
3.
4. $username = $_POST['username'];
5. $password = $_POST['password'];
6.
7. $callback = greeting($username, $password);
8.
9. echo $callback;
10.
11.  ?>
```

controller

```
1. <?php
2. function greeting($user, $pass){
3.     $user = strtolower(trim($user));
4.     $pass = trim($pass);
5.     // validate user if Alireza with Al88--
6.     if($user == "alireza" && $pass == "Al88--")
7.         return "Hello "."$user". "\n"."Welcome back "."$user";
8.     else
9.         return "Unknown user plz go away..";
10.  }
```

print_r(array)

از این دستور زمانی استفاده میکنیم که بخواهیم یک پرینت قابل خواندن تمام نوع متغیر ها داشته باشیم، یعنی حتی با این دستور میتوان برخلاف دستور print، تمام دیتا های آرایه ها را مشاهده کنیم.

var_dump(array)

دستور دیگری به نام ور_دامپ وجود دارد که میتوان با استفاده از آن مانند دستور print_r تمام محتوایات آرایه را بدون هیچ Loop event مشاهده کرد، تفاوت آن با print_r این است که، var_dump علاوه بر مقادیر، نوع و طول هر کدام را بر میگرداند.

array_value(array)

آرایه را به مقدار خالص خود بر می گرداند، یعنی از آن حالت key: value به حالت numerical باز میگردد.

associative array

آرایه ای است که مانند دیکشنری پایتون عمل میکند، یعنی داده ها را به صورت کلید: مقدار ذخیره میکند.

حلقه ها:

حلقه فور، for:

```
1. for ($i = 0; $i<10; $i++){
2.         echo "Number $i";
3. }
```

حلقه وایل یا، while:

یک دستور ساده برای داشتن یک حلقه است که دارای شرط بقاست و افزاینده ای ندارد و ما باید خودمان افزایند را به صورت گلوبال تعریف کنیم و در درون حلقه به آن اضافه کنیم که حلقه فایل ما تبدیل به حلقه بی نهایت نشود.

```
1. $index = 0;
2. while($index < 10){
3.         echo "$index";
4.      $index ++;
5. }
```

foreach

دستور سریع تر و بهینه برای نمایش مقادیر آرایه ها میباشد، که کار با آن خیلی راحت است، سینتکس آن به شکل زیر است:

```
1. foreach (<<your array>> as $item){
2.     echo $item;
3. }
```

count(array)

با استفاده از این دستور میتوان اندازه یک آرایه را بررسی کرد.

استفاده از Switch case:

```
1. $n = 45;
2.
3. switch ($n) {
4.     case gettype($n) == "String":
5.         echo "String";
6.         break;
7.     case  gettype($n) == "integer":
8.         echo "Int";
9.         break;
10.          default:
11.              echo "Nan";
12.  }
13.
```

استفاده از کلاس

کلاس در PHP به صورت زیر نوشته می شود:

```
1. class MyClass{
2.     // your properties
3.    // your methods
4. }
```

تابع construct__

دلیل اصلی استفاده ما از این تابع آن است که بتوانیم تمام پراپرتی هایمان را از طریق خود کلاس به مانند تابعی با آرگومان های معنا دار و مختلف بنویسیم و وارد کلاس مربوطه کنیم.

4 قاعده اصلی در کلاس ها

Abstraction

زمانی که ما کلاس خود را ایجاد کرده ایم شاید بخواهیم در برخی از پراپرتی ها در مورد پراپرتی های پرایویت صحبت کنیم به همین خاطر میتوانیم یک کلاس دیگیری برای آن ایجاد کنیم و آن پراپرتی را به صورت پرایویت تعریف کنیم که خود آن پراپرتی از کلاس اول نیز استفاده میکند و ما میتوانید در برنامه اصلی برای دسترسی به آن، در کلاس دوم توابع مورد نظر را بنویسیم که برایمان آن پراپرتی را برگرداند.

Encapsulation

به طور کلی زمانی کپسوله سازی بین کلاس ها را انجام میدهیم که بخواهیم دو یا چند کلاس را از هم جدا کنیم که هر کدام وظیفه مخصوص به خودشان را انجام بدهند.

inheritance

در رابطه با ارث بری میتوان گفت که همان شئ اصلی با پراپرتی های مخصوص به خود است، یعنی آنکه از پراپرتی های کلاس اصلی استفاده میکنید + پراپرتی هایی که خودش دارد میتواند نقشی در برنامه ایفا کند برای مثال:

```
1. class Car {
2.       $seatNumber = 5;
3.      $doorNumber = 4;
4.      $name;
5. }
6. class Electronic extended Car{
7.       $butteryLevel = 58;
8.      function recharge(){
9.        $butteryLevel = 100;
10.       }
11.  }
12.  $MyCar = new $Car();
13.  $MyTesla = new $Electronic();
14.  $MyTesla->recharge();
```

Polymorphism

زمانی از چند ریختی استفاده میکنیم که بخواهیم از یک کلاس چندین حالات را داشته باشیم اینطور از نوشته شدن حالات مختلف با کلاس ها و ابجکت های مختلف جلوگیری میشود.

نکات مهم:

زمانی که ما از include برای وارد کردن فایل سورس خود استفاده میکنیم زمانی که آن فایل وجود نداشته باشد ما در برنامه خود به هشدار هایی برخواهیم خورد و برنامه بی توقف ادامه خواهد یافت.

زمانی که از Require برای اضافه کردن فایل سورس خود به برنامه استفاده میکنیم در صورتی که آن فایل وجود نداشته باشد به ارور فتال برخواهیم خورد و برنامه متوقف خواهد شد.

```
1. include './controller.php';
2. require './controller.php';
```

تفاوت پراپرتی های Protected و Private و Public

زمانی از متغیر های protected در کلاس یا متدها و پراپرتی های خود استفاده میکنیم که میخواهیم آن متد یا پراپرتی در تمامی کلاس ها یعنی چه کلاس والد چه کلاس بچه ها، در همه جا در کلاس های جاری قابل استفاده باشند.

زمانی از Private استفاده میکنیم که آن پراپتی یا فانکشن فقط در آن کلاس مورد استفاده قرار گیرد نه در کلاس ها و بچه های دیگر.

زمانی از Public استفاده میکنیم که بخواهیم پراپرتی ما به صورت کلی در همه جا قابل استفاده باشد.

مراجع

روش های برنامه‌نویسی: https://cs.lmu.edu/~ray/notes/paradigms