

برنامه نویسی مبتنی بر وب

استاد: برنجی

علیرضا سلطانی نشان

ترم سوم

1399.07.15

فهرست مطالب

3	تفاوت بین Script Language و Programming Language
3	دسته بندیها
3	تفسیر
4	توسعه
4	سرعت
4	منظور از شئگرایی چیست؟
6	فریمورک
6	برخی از مزایای استفاده از فریمورک ها:
6	Editor و IDE، تفاوت در چیست؟
7	انواع روش های برنامه نویسی
7	برنامه نویسی ضروری
8	برنامه نویسی ساختارمند
8	برنامه نویسی تابعی (عملکردی)
10	برنامه نویسی اظهاری
10	تاریخچه PHP
12	نوشتن کد در زبان PHP
12	دستور echo
12	دستور Print

- 13 تفاوت echo با print
- 13 متغیرها
- 13 ثابت ها
- 14 Double quote و Single quote تفاوت
- 15 مراجع

تفاوت بین Script Language و Programming Language

مهم ترین تفاوت بین زبان برنامه نویسی با زبان اسکریپت نویسی، به نوع اجرای¹ آنها بر می‌گردد. در Programming Language یک مترجم² وجود دارد تا برنامه نوشته شده شما را از زبان سطح بالای³ برنامه نویسی به زبان سطح پایین ماشین⁴، تبدیل کند. در زبان اسکریپت نویسی، دیگر خبری از یک مترجم نیست، بلکه یک مفسر⁵ 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 کاملاً 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 بندی‌ها⁶

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 کدهای جی اس را اجرا می‌کند در واقع همان مرورگر است، یا مثلاً در Node.js برای اجرای بلادرنگ⁹ 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.

¹ Execute

² Compiler

³ High Level

⁴ Low Level (Machine Level)

⁵ Interpreter

⁶ Categories

⁹ RealTime

توسعه¹⁰

نوشتن کد با استفاده یک زبان برنامه نویسی مانند سوئیفت یا سی، نسبتاً 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 می کنند اگر مشکلی یا خطایی در فرآیند اجرای برنامه وجود داشته باشد، برنامه همان اولین زمان متوقف می شوند، 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 می شوند، اگر در یکی از خطها مشکلی باشد کل برنامه متوقف می شود و نتیجه ناقصی را خواهیم داشت.

منظور از شیء‌گرایی¹¹ چیست؟

برنامه نویسی شیء‌گرا یکی از روش های برنامه نویسی است که بر پایه یکسری از مفاهیم کلاس ها و شیء ها است. شیء‌گرایی 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 می کنیم و می توانیم نمونه های¹² زیاد و مختلفی از آن ها داشته باشیم. به خاطر

¹⁰ Development

¹¹ Object Oriented Programming

¹² Instances

اینکه شی‌گرایی یک شیوه برنامه نویسی است ما در این زمینه زبان های زیادی مانند، سی ها، پایتون، جی اس و غیره را داریم.

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 شی ها از زبان JS استفاده کردم:

```
1. } const users = {
2.   name: 'Alireza',
3.   family: 'Soltani',
4.   age: getAge = (userBirthday) => {
5.     const realAge = Date.now() - userBirthday
6.     return realAge
7.   },
8.   field: "JS programmer"

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

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

```
1. class users {
2.   constructor(name, family, birthday) {
3.     this.name = name
4.     this.family = family
5.     this.birthday = birthday
6.   }
7.
8.   calculateUsers() {
9.     const years = Date.now() - this.birthday
10.    const name = this.name
11.    const family = this.family
12.    return {
13.      name,
14.      family,
15.      years
16.    }
17.  }
18. }
19.
20. const user1 = new users('Alireza', 'Soltani', 2000)
21. const alirezaSoltaniNeshan = user1.calculateUsers()
22.
23. console.log(alirezaSoltaniNeshan)
24.
```

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 فریم‌ورک‌های زیادی در استفاده‌های 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 فریم‌ورک‌ها:

- کاهش زمان کد زدن
- افزایش بهره‌وری و صرفه جویی در تعداد خط‌ها
- 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 فریم‌ورک‌ها نصب خواهند شد که ممکن است در ابتدای نصب نیازمند به اینترنت باشید که تمامی پکیج‌ها و برنامه‌های 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 پکیج‌ها 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، (کاملاً یک ویرایش کننده حالا هر چیزی) برای مثال، شما می‌توانید برنامه‌ای با هر زبانی بنویسید، به شرط آنکه تمام ملزومات و کتابخانه‌های آن نصب باشد، به عنوان پیش‌فرض VS code، برای نوشتن یک برنامه فلاتر نیازمند نصب تمامی پکیج‌ها در هر پروژه خود دارید، یا مثلاً در هنگام برنامه‌نویسی سرور باید در زبان جی‌اس ابتدا نیاز به بسته‌های node_modules در هر پروژه دارید، که بایستی به وسیله پکیج‌منیجر npm کتابخانه‌ها و فریم‌ورک‌های موردنظر را نصب کنید. این نصب و به اصطلاح init کردن در هنگام استفاده از ویرایش به صورت یکپارچه برای همیشگی نیست بلکه هر بار پروژه زدن نیاز به دوباره کاری داریم، اما در IDE مانند Xcode هنگام پروژه زدن یا init کردن برنامه خود همه وابستگی‌ها، همان ابتدای نصب مشخص شده‌اند و نیازی به دوباره کاری نیست.

انواع روش‌های برنامه‌نویسی

روش‌های برنامه‌نویسی، یعنی چه راه‌هایی برای نوشتن یک برنامه وجود دارد.

روش‌های برنامه‌نویسی هم به تعداد هستند: که در اینجا 4 نمونه از آشناترین آنها را بررسی خواهیم کرد.

برنامه‌نویسی ضروری¹³

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 ساختارمند¹⁴

برنامه‌نویسی ساختارمند هم از هم‌نوع برنامه‌نویسی ضروری می‌باشد با این تفاوت که کنترل جریان به صورت حلقه‌های 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);
```

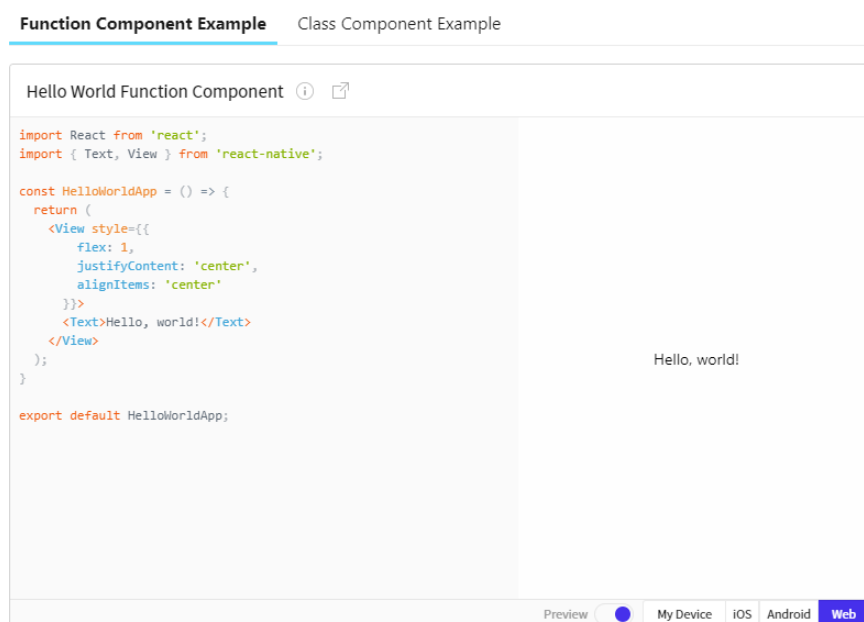
برنامه‌نویسی تابعی (عملکردی)¹⁵

در برنامه‌نویسی تابعی تمام عملیات درون یک تابع نوشته می‌شود، در این یک تابع ممکن است تعداد کم یا زیادی از توابع برای انجام یکسری Task ها وجود داشته باشد. (یک مثال) در زمانی که

¹⁴ Structed

¹⁵ Functional Programming

شروع به یادگیری برنامه نویسی جی اس کردم و زمانی که خواستم بروی فریمورک ریکت نیتو مشغول شوم، دو راه برای نوشتن برنامه های React Native با JS وجود داشت، یک برنامه نویسی extended Class (حالی نه مثل polymorphism) و دیگری برنامه نویسی به صورت Functional. تا قبل از سال 2018 برنامه نویسی با فریمورک React Native به صورت تابعی، در حقیقت برای نمایش محتوای ثابت¹⁶ بود 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 می توانید به یک نتیجه برسید.



شکل 11 اکسیو کامپایلر، روش تابعی یا کلاسی

¹⁶ Static Contents

برنامه‌نویسی اظهاری¹⁷

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 معمولاً توسط یک سرویس دهنده وب مانند (درحالت لوکال) Wamp، توسط مفسر PHP پردازش می‌شوند 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 مستقل¹⁸، یا کنترل ربات ها یا برنده های کنترلی. و همچنین مانند پلتفرم نود جی اس می‌تواند به صورت خط فرمان¹⁹ 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 تقریباً 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 تقریباً 18 ماه پس از اعلام رسمی در مورد ازسر گیری فعالیت های برنامه نویسی برای نسخه جدید، PHP 4.0 به بیرون منتشر شد. بسیاری از مردم، نسخه 4 را جنجالی ترین نسخه از این زبان می دانستند و در بسیاری از فروم ها صحبت هایی در مورد آن میشد. بعد از چند ماه، سایت netcraf گزارشی بیرون داد که تخمین می زد 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، رابط ها و ارتقای طریقه مدیریت کردن اشیاء.

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 نسخه ششم این زبان، ویژگی هایی زیادی معرفی شدند که در اینجا به چندتا از آنها اشاره می کنم:

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 Promise ها و غیره.

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 پردازید.

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 همچنین 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. ?>
```

متغیرها قسمتی‌هایی 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.

انواع مختلفی که حتما می‌دانیم وجود دارد:

رشته‌ها، اعداد (اعشاری و صحیح)، آرایه‌ها و آبجکت‌ها فانکشن‌ها و غیره.

ثابت‌ها

زمانی که از ثابت‌ها 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 Double quote و Single 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 خواهیم کرد

مراجع

روش های برنامه نویسی: <https://cs.lmu.edu/~ray/notes/paradigms>