زبان برنامه نویسی C# و فریم ورک Blazor



سی شارپ (C#) یک زبان برنامه‌نویسی شیءگرا است که توسط شرکت مایکروسافت توسعه داده شده است. این زبان برای توسعه برنامه‌ها در محیط دات‌نت (Microsoft .NET) و بستر برنامه‌های ویندوز استفاده می‌شود.C# از زبان‌های برنامه‌نویسی C و C++ الهام گرفته و به طور واضح تحت تأثیر زبان Java قرار دارد.

تاریخچه C# به سال ۱۹۹۹ بازمی‌گردد که مایکروسافت آن را برای توسعه برنامه‌هایی بر روی پلتفرم NET. معرفی کرد. نام C# به خاطر شباهت نحوی و ساختاری با زبان C++ انتخاب شد. این زبان در ابتدا به عنوان بخشی از فعالیت‌های تحقیق و توسعه شرکت مایکروسافت در زمینه تحقق بینایی مصنوعی و رایانش توزیع‌شده معرفی شد. در سال ۲۰۰۲، نسخه اولیه از C# به همراه فریم‌ورک NET. منتشر شد.

در تعریفی دیگر، اگر بخواهیم بگوییم که سی شارپ چیست، باید بگوییم زبانی قوی و انعطاف‌پذیر است که از آن برای توسعه برنامه‌های متنوعی استفاده می‌شود. به عنوان یک زبان شیءگرا، C# از اصول مفهومی مانند کلاس‌ها، اشیاء، ارث‌بری، چندریختی و استثناءها پشتیبانی می‌کند. همچنین، C# قابلیت‌های بسیاری برای برنامه‌نویسی وب، توسعه نرم‌افزارهای سمت سرور و کلاینت، نرم‌افزارهای تحت وب و برنامه‌نویسی بازی‌ها را فراهم می‌کند.

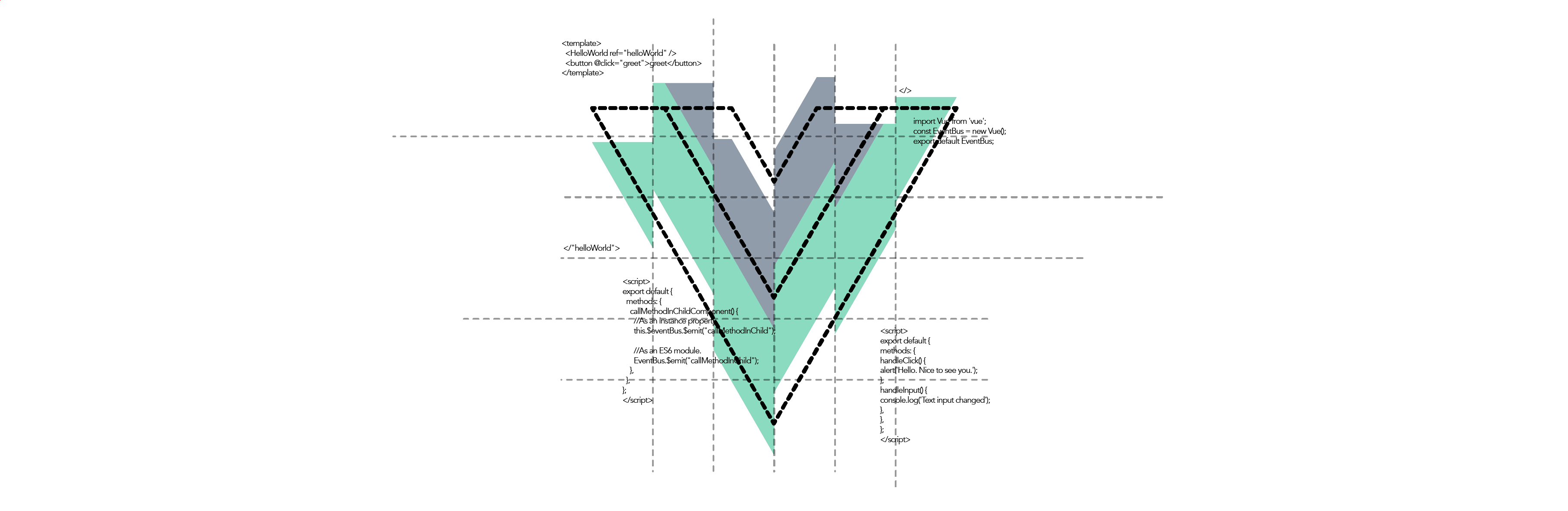
اگر بخواهیم در مورد کاربرد سی شارپ به صورت خلاصه توضیح دهیم، باید بگوییم که سی شارپ در حال حاضر به عنوان زبان اصلی برنامه‌نویسی برای توسعه نرم‌افزارهای ویندوز، برنامه‌های دسکتاپ، برنامه‌های موبایل با استفاده از Xamarin و توسعه برنامه‌های وب با استفاده از ASP.NET استفاده می‌شود. زبان C# توسط جامعه برنامه‌نویسی به خوبی پذیرفته شده است و منابع زیادی برای یادگیری و توسعه با C# در دسترس است.

تیم Asp.net مجموعه ابزاری را به نام Blazor ساخته اند که یک فریمورک سمت کلاینت

(client side) برای برنامه های دات نت است. تفاوت این فریمورک با فریمورک های سمت کلاینت دیگر مانند angular, react این است که anuglar و react هنگامی که برای مرورگرف کامپایل می شوند به کد جاوا اسکریپت که قابل درک برای مرورگر ها است تبدیل می شوند. اما blazor به کد جاوااسکریپت تبدیل نمی شود و به وب اسمبلی کامپایل می شود. از ویژگی هایی که این فریمورک دارد موارد زیر است:

* پایه آن تکنولوژی های امروزی مثل html , css است با این تفاوت که به جای استفاده از جاوا اسکریپت از سی شارپ و دستورات Razor استفاده می کنید.
* شما با استفاده از این تکنولوژي می توانید همه قابلیت های پلتفرم SPA(برنامه های تک صفحه ای وب) را در دسترس دارید.
* این فریمورک از WebAssembly استفاده می کند.

زبان برنامه نویسیJava Script و فریمورک Vue.Js



جاوااسکریپت (JavaScript) زبان برنامه‌نویسی سطح بالاو پویا مبتنی بر شی است. از JS در کنار  HTML و CSS، به عنوان یکی از سه هسته تشکیل دهنده صفحات وب، یاد می‌شود. البته استفااده از JS فقط به سمت کاربر (Front-end) خلاصه نمی‌شود و امروزه می‌توان با فریمورک‌هایی مانند vue.js  برنامه نویسی سمت سرور (Back-end) نیز انجام داد. پس می‌توان جاوااسکریپت(JS) را یک زبان برنامه‌نویسی Both Side دانست.  
بر خلاف تشابه اسمی دو زبان برنامه نویسی جاوا (Java) و جاوااسکریپت (JavaScript) و باور عده‌ای که هر دو را یک زبان قلمداد می‌کنند، این دو هیچ‌ ارتباطی با یکدیگر، جز تشابه اسمی، ندارند. ساختار جاوااسکریپت شباهت زیادی به جاوا و C++ دارد. جاوااسکریپت زبان برنامه‌نویسی شی‌گرایی (Object-Oriented) و ساخت یافته (Structured) است. با این زبان می توان به محتوای داخل html دسترسی پیدا کرد.

ویو یک فریم‌ورک متن باز زبان javascript است که از آن برای توسعه رابط کاربری استفاده می­‌شود. Vue.js برای اولین بار در دسامبر سال 2013 توسط Evan You، یکی از کارمندان سابق گوگل، به دنیا معرفی شد. ایوان که در شغل پیشین خود از فریم­ورک Angular js در توسعه برنامه‌­ها استفاده می‌کرد، تصمیم گرفت تا فریم‌ورکی برای استفاده در پروژه‌­های خود خلق کند که به اندازه Angular قدرتمند و در عین حال سبک‌وزن و انعطاف‌پذیر باشد. شما می­‌توانید از Vue.js در ساخت اپلیکیشن­‌های تحت وب تک صفحه‌ای یا حتی وب‌سایت­‌هایی در مقیاس بزرگ با ساختارهای مختلف استفاده کنید. همچنین می‌­توانید کامپوننت‌­هایی ایجاد کرده و در سایر بخش‌های پروژه یا حتی پروژه­‌های دیگر از آن استفاده کنید.

زبان برنامه نویسی TypeScript و فریمورک Angular



تایپ اسکریپت به انگلیسی: (TypeScript) یک زبان [**برنامه‌نویسی**](https://7learn.com/blog/learning-programming) چندسکویی، متن باز و کامپایلری است که توسط شرکت مایکروسافت توسعه داده شده و پشتیبانی می‌شود. تایپ اسکریپت یکی از زیر مجموعه‌های زبان جاوا اسکریپت است که به شما کمک می‌کند تا کدهای خود را به روش ساده‌تری نوشته و سپس با کامپایل‌ کردن آن‌ها خروجی استاندارد کد جاوا اسکریپت را به دست آورید. امکانات پیشرفته‌ای که مایکروسافت به تایپ اسکریپت افزوده است، آن را به زبان‌هایی مانند #C، جاوا و... نزدیک ساخته است. تایپ اسکریپت برای توسعه‌ی پروژه‌های بزرگ جاوا اسکریپت طراحی شده است و فرآیند توسعه را تا حد زیادی ساده می‌کند. امروزه تایپ اسکریپت به یکی از محبوب‌ترین زبان‌های دنیا تبدیل شده است و روز به روز افراد بیش‌تری را به خود جذب می‌کند. اما ممکن است این سوال برای شما نیز پیش آید که چرا باید تایپ اسکریپت را یاد بگیریم و استفاده و یادگیری آن چه مزایایی به همراه دارد؟ در ادامه‌ی این مقاله به این سوالات پاسخ خواهیم داد و زبان تایپ اسکریپت و مزایای آن را به شما معرفی خواهیم کرد، با ما همراه باشید.

انگولار (Angular)‌ یکی از فریم‌ورک‌های مبتنی بر TypeScript است که به صورت رایگان و متن-باز، توسط تیم Angular در Google توسعه یافته است. این فریمورک محبوبیت خاصی بین توسعه‌دهندگان دارد و هم از سمت گوگل و هم از سمت جامعه توسعه‌دهندگان انگولار پشتیبانی می‌شود. از مهم‌ترین کاربردهای انگولار، توسعه وب اپلیکیشن‌های تک صفحه‌ای (Single page) و وب‌سایت‌های تعاملی است.

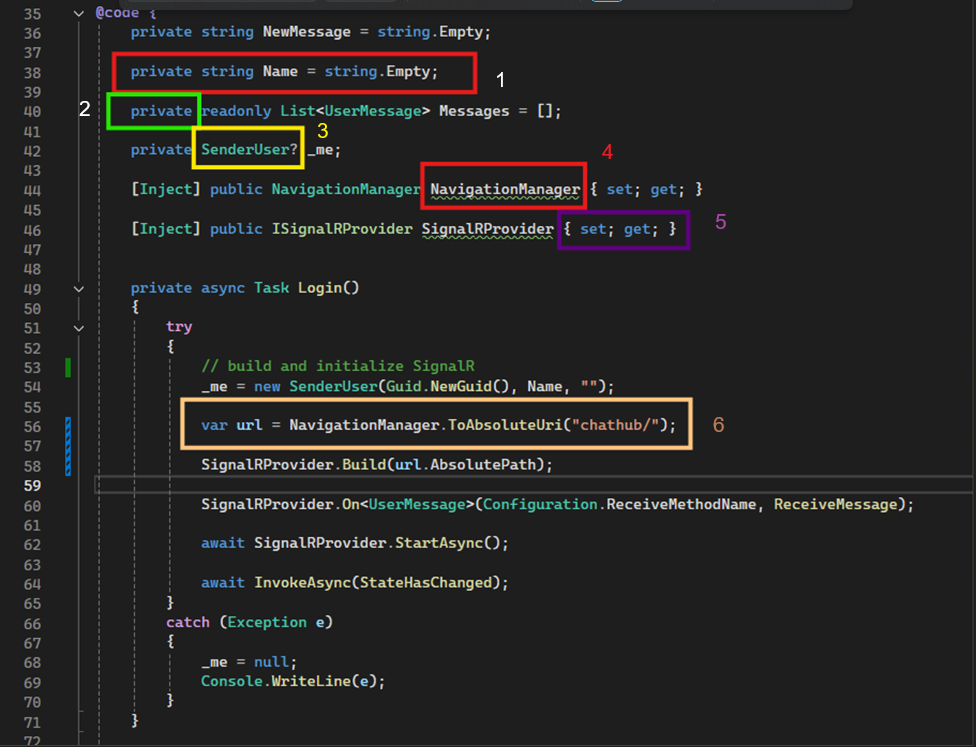
فریم ورک انگولار ویژگی‌هایی دارد که آن را برای توسعه اپلیکیشن‌های وب ایده‌آل می‌کند. این ویژگی‌ها عبارتند از:

* چارچوب مبتنی بر مولفه (Component-based) که در ساخت اپلیکیشن‌های مقیاس‌پذیر به افراد کمک می‌کند.
* دارای انواع کتابخانه‌های یکپارچه که برای افزودن ویژگی‌های مختلف مانند فرم‌ها، ارتباط مشتریان با سرور و … به کمک توسعه‌دهندگان می‌آیند.
* دارای مجموعه بزرگی از ابزارهای توسعه که در فرایند توسعه نرم‌افزار، از مراحل ابتدایی تا توسعه، بیلد، تست و به‌روزرسانی به کار می‌آیند.

|  |  |  |  |
| --- | --- | --- | --- |
| ویژگی | C# | JavaScript (JS) | TypeScript (TS) |
| نوع زبان | زبان شیءگرا و چندمنظوره | زبان اسکریپت‌نویسی و تفسیری | زبان اسکریپت‌نویسی با تایپ قوی |
| محیط اجرا | معمولاً بر روی .NET Framework یا Core | مرورگرها و Node.js | مرورگرها، Node.js (پس از کامپایل به JS) |
| تایپ داده‌ها | ایستا (Static) | پویا (Dynamic) | ایستا (Static) |
| کاربرد اصلی | برنامه‌نویسی دسکتاپ، وب و موبایل | توسعه وب، اپلیکیشن‌های کوچک | توسعه وب با تاکید بر کدنویسی ایمن‌تر |
| پشتیبانی از شیءگرایی | کامل و ساختارمند | غیرساختارمند (اما امکان‌پذیر) | ساختارمند و مشابه C# |
| نیاز به کامپایل | بله | خیر | بله (کامپایل به JavaScript) |
| عملکرد در اجرا | بالا (وابسته به پلتفرم .NET) | متغیر (وابسته به مرورگر یا موتور JS) | مشابه JavaScript (بعد از کامپایل) |
| قابلیت‌های مدرن | پشتیبانی کامل از ویژگی‌های جدید | بله | بله، بهبود یافته |
| یادگیری و جامعه | مناسب برای توسعه‌دهندگان حرفه‌ای | ساده برای مبتدیان | مناسب برای توسعه‌دهندگان متوسط به بالا |
| ابزارهای کمکی | Visual Studio | هر ویرایشگر متن یا IDE | Visual Studio Code و دیگر IDEها |

|  |  |  |  |
| --- | --- | --- | --- |
| **ویژگی‌ها** | **Vue.js** | **Angular** | **Blazor** |
| **زبان پایه** | جاوااسکریپت/تایپ‌اسکریپت | تایپ‌اسکریپت | سی‌شارپ |
| **مدل رندرینگ** | کلاینت‌محور (SPA) | کلاینت‌محور و SSR | کلاینت‌محور (WebAssembly) یا SSR |
| **منحنی یادگیری** | آسان | متوسط به سخت | متوسط |
| **سرعت توسعه** | سریع و سبک | کندتر به دلیل پیچیدگی‌ها | وابسته به تجربه توسعه‌دهنده C# |
| **اندازه بسته** | کوچک | بزرگ‌تر | متوسط |
| **کاربردها** | پروژه‌های سبک و متوسط | پروژه‌های بزرگ و سازمانی | اپلیکیشن‌های متصل به دات‌نت |
| **وابستگی‌ها** | حداقل | زیاد | اکوسیستم دات‌نت |
| **عملکرد** | سریع و سبک | عملکرد مناسب اما سنگین‌تر | عملکرد بالا (با WebAssembly) |
| **پشتیبانی از موبایل** | با ابزارهایی مثل Quasar | Angular Native | Maui Blazor (در حال توسعه) |
| **جامعه کاربری** | بزرگ و فعال | بسیار بزرگ و پایدار | در حال رشد |
| **ابزار توسعه** | Vue CLI، Vite | Angular CLI | Visual Studio، Blazor Server |

تعریف متغیر :

C# :

در c# میتوان مانند شکل 1 در سطح کلاس متغییر تعریف کرد .

شکل 2 سطح دسترسی متغیر را نمایش میدهد که میتواند (Public – Internal – Protected - Private)

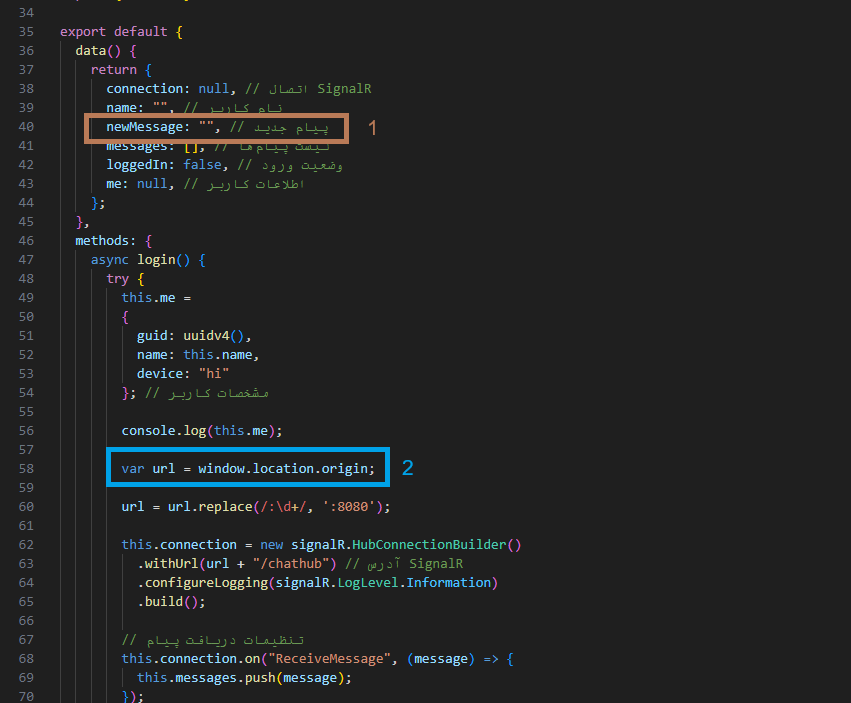
شکل 3 تایین تایپ متغیر را نشان میدهد.

شکل 4 اسم متغییر است.

شکل 5 برای تعیین دسترسی مقدار دهی و گرفتن مقدار استفاده میشود (مثال : متغییر میتواند از تمامی کلاس ها دیده شود اما تنها از درون همان کلاس بتواند مقدار آن تغییر کند)

شکل 6 تعریف یک متغییر محلی به نام url را نشان میدهد و تنها درون همین متد قابل دسترس است.

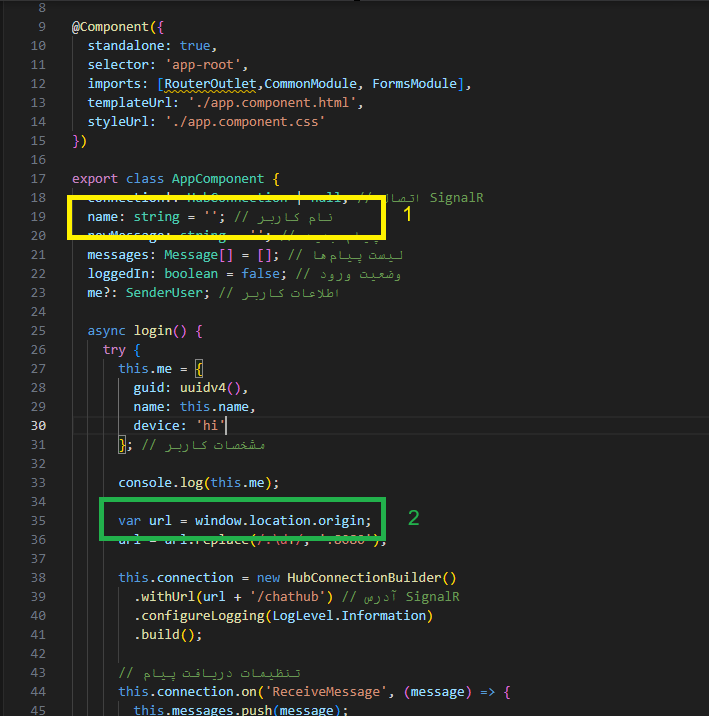
Var به این معنی است که کامپایلر بر اساس مقداری که درون آن ریخته شود به تایپ آن را خواهد فهمید.

Js :

تعریف متغیر در سطح فایل در Vue.js به شکل 1 است و باید درون Return نوشته شود .

تعریف متغیر محلی هم یه صورت 2 است , و میتوان از آن در خود تابع استفاده کرد.

در js مفهومی به اسم کلاس یا سطح دسترسی نداریم

TS :

در Ts متغیر ها اول کلاس و به فرم (تایپ متعیر : نام) (شکل 1) تاریف میشوند.

متغیر های محلی نیز به صورت زیر است.

میتوان از سطوح دسترسی (Public Protected private) نیز استفاده کرد .

دستورات شرط و حلقه در هر سه زبان به یک صورت است .

**تعریف کلاس :**

C# :

A screenshot of a computer program

Description automatically generated

در c# کلاس ها با کلید واژه class و یک سطح دسترسی تعریف میشوند .

تعریف متغییر نیز از قبل گفته شده است.

TS :



در تایپ اسکریپت کلاس ها به صورت export و با کلید واژه Class تعریف میشود

JS : از کلاس پشتیبانی نمیکند.

توابع :

C# :

A screen shot of a computer

Description automatically generated

توابع در سی شارپ به فرمت

( نام متغیر تایپ متغیر )نام تابع تایپ خروجی سطح دسترسی

تعریف میشوند.

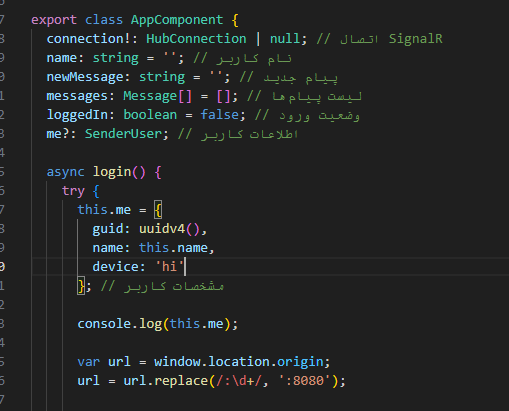
Vue.js :

A screen shot of a computer program

Description automatically generated

در Vue توابع درون متغییر methods تعریف شده و ورودی ها نیز درون پرانتز نوشته میشود .  
async کلید واژه ای برای تایین اجرای نا همزمان است تا Dom از کار نیفتد.

Ts :



در تایپ اسکریپت نیز شبیه js است با این تفاوت که مانند c# درون کلاس ها تعریف میشوند .

**1.زبان پایه و فناوری**

* **Vue.js**: مبتنی بر **JavaScript** و **TypeScript** است و برای توسعه رابط کاربری سبک و مدرن طراحی شده.
* **Angular**: به طور کامل بر اساس **TypeScript** پیاده‌سازی شده و برای پروژه‌های سازمانی و پیچیده مناسب است.
* **Blazor**: بخشی از اکوسیستم دات‌نت است و از **C#** برای توسعه استفاده می‌کند. این ابزار امکان اجرای کد سی‌شارپ در مرورگر با استفاده از **WebAssembly** را فراهم می‌کند.

**2. معماری و مدل رندرینگ**

* **Vue.js**: معماری مبتنی بر کامپوننت است و می‌تواند به صورت **کلاینت‌محور (SPA)** یا **SSR (Server-Side Rendering)** اجرا شود.
* **Angular**: معماری کامپوننتی و ماژولار دارد. از **دوطرفه‌سازی داده‌ها** (Two-Way Data Binding) و **Dependency Injection** استفاده می‌کند.
* **Blazor**: می‌تواند به دو شکل **کلاینت‌محور (WebAssembly)** یا **سرورمحور (Blazor Server)** اجرا شود. از مدل **یکپارچه با دات‌نت** استفاده می‌کند.

**3. منحنی یادگیری**

* **Vue.js**: یادگیری آسان با مستندات ساده. مناسب برای توسعه‌دهندگان مبتدی و متوسط.
* **Angular**: منحنی یادگیری بالاتر به دلیل پیچیدگی‌ها، اما ابزارها و ویژگی‌های غنی آن در پروژه‌های بزرگ کاربردی است.
* **Blazor**: مناسب برای توسعه‌دهندگانی که با **C#** و اکوسیستم دات‌نت آشنایی دارند.

**4. کاربردها و بازار هدف**

* **Vue.js**: مناسب برای پروژه‌های سبک تا متوسط، مانند وب‌سایت‌های تعاملی یا داشبوردهای کوچک.
* **Angular**: برای اپلیکیشن‌های بزرگ و پیچیده سازمانی که نیاز به پشتیبانی از تیم‌های بزرگ توسعه دارند.
* **Blazor**: بهترین گزینه برای شرکت‌هایی که از فناوری دات‌نت استفاده می‌کنند یا نیاز به استفاده مجدد از کدهای بک‌اند در فرانت‌اند دارند.

**5. عملکرد و مقیاس‌پذیری**

* **Vue.js**: سبک و سریع، با تمرکز بر حداقل وابستگی‌ها.
* **Angular**: به دلیل ویژگی‌های غنی و ماژولار، نسبت به Vue.js سنگین‌تر است اما مقیاس‌پذیری خوبی دارد.
* **Blazor**: عملکرد خوبی با استفاده از WebAssembly ارائه می‌دهد، اما اجرای اولیه ممکن است به دلیل دانلود فایل‌های WebAssembly کمی کند باشد.

**6. ابزارها و جامعه**

* **Vue.js**: ابزارهایی مثل **Vue CLI** و **Vite** با جامعه کاربری بسیار بزرگ و فعال.
* **Angular**: ابزارهای توسعه قدرتمندی مانند **Angular CLI** و جامعه‌ای پایدار و باسابقه.
* **Blazor**: ابزارهایی مانند **Visual Studio** و **.NET CLI**، با جامعه‌ای که در حال رشد است.

**7. پشتیبانی از موبایل**

* **Vue.js**: می‌توان از فریم‌ورک‌هایی مانند **Quasar** برای توسعه اپلیکیشن‌های موبایل استفاده کرد.
* **Angular**: امکان توسعه اپلیکیشن‌های موبایل با استفاده از **Ionic Framework**.
* **Blazor**: با ابزارهایی مثل **MAUI Blazor**، می‌توان اپلیکیشن‌های موبایل ایجاد کرد.

**1. Vue.js**

* **مدل انقیاد:**
  + پشتیبانی از انقیاد یک‌طرفه (**One-Way Data Binding**) و دوطرفه (**Two-Way Data Binding**).
  + استفاده از دستور v-model برای انقیاد دوطرفه در فرم‌ها و ورودی‌ها.
* **انعطاف‌پذیری:**
  + Vue بسیار انعطاف‌پذیر است و به توسعه‌دهنده این امکان را می‌دهد که بین انقیاد یک‌طرفه و دوطرفه انتخاب کند.
* **مزایا:**
  + عملکرد سریع‌تر با انقیاد یک‌طرفه.
  + سادگی و راحتی انقیاد دوطرفه برای مدیریت فرم‌ها.

**2. Angular**

* **مدل انقیاد:**
  + انقیاد داده‌ها در Angular به صورت پیش‌فرض **دوطرفه (Two-Way Data Binding)** است که از طریق دستور [(ngModel)] پیاده‌سازی می‌شود.
  + همچنین، پشتیبانی از انقیاد یک‌طرفه (**One-Way Data Binding**) در قالب‌های Angular وجود دارد.
* **انعطاف‌پذیری:**
  + Angular امکان تعریف دقیق نحوه تعامل داده‌ها بین **View** و **Model** را فراهم می‌کند.
* **مزایا:**
  + انقیاد دوطرفه باعث کاهش پیچیدگی کد در فرم‌های پیچیده می‌شود.
  + انقیاد یک‌طرفه می‌تواند در بهینه‌سازی عملکرد و ردیابی تغییرات داده‌ها کمک کند.

**3. Blazor**

* **مدل انقیاد:**
  + Blazor پشتیبانی قوی از انقیاد دوطرفه (**Two-Way Data Binding**) دارد و این ویژگی با استفاده از دستور @bind انجام می‌شود.
  + امکان انقیاد یک‌طرفه نیز برای نمایش داده‌های فقط خواندنی وجود دارد.
* **انعطاف‌پذیری:**
  + انقیاد داده در Blazor به طور عمیق با **C#** و دات‌نت ادغام شده است، که استفاده از داده‌ها و مدل‌ها را در هر دو سمت کلاینت و سرور ساده می‌کند.
* **مزایا:**
  + با توجه به اینکه زبان اصلی Blazor سی‌شارپ است، انقیاد دوطرفه می‌تواند به سادگی داده‌های پیچیده را مدیریت کند.
  + امکان استفاده مجدد از منطق بک‌اند در فرانت‌اند.

|  |  |  |  |
| --- | --- | --- | --- |
| **ویژگی** | **Vue.js** | **Angular** | **Blazor** |
| **انقیاد یک‌طرفه** | بله | بله | بله |
| **انقیاد دوطرفه** | بله (با v-model) | بله (با [(ngModel)]) | بله (با @bind) |
| **پیاده‌سازی پیش‌فرض** | یک‌طرفه | دوطرفه | دوطرفه |
| **پیچیدگی انقیاد** | ساده | متوسط | ساده |
| **انعطاف‌پذیری** | بالا | بالا | متوسط به بالا |

کد برنامه پیام رسان با هر سه زبان و فریم ورک در گیت هاب به آدرس :  
<https://github.com/mosha99/Messenger>