زبان **golang**

فهرست

[معرفی زبان go 2](#_Toc27433119)

[تاریخچه زبان go 3](#_Toc27433120)

[کاربرد های زبان go 3](#_Toc27433121)

[برنامه نویسی میکرو سرویس‌ها و سرورها 3](#_Toc27433122)

[برنامه نویسی سیستمی 3](#_Toc27433123)

[برنامه نویسی همروند 4](#_Toc27433124)

[مفسری یا کامپایلری؟ 4](#_Toc27433125)

[ویژگی های مترجم 4](#_Toc27433126)

[انواع انقیادها و زمان های انقیاد 4](#_Toc27433127)

[انقیادهای زمان طراحی زبان 4](#_Toc27433128)

[انقیادهای زمان پیاده سازی 4](#_Toc27433129)

[انقیادهای زمان ترجمه 4](#_Toc27433130)

[انقیادهای زمان اجرا 4](#_Toc27433131)

[عناصر نحوی زبان 4](#_Toc27433132)

[مجموعه‌ی کارکترها 4](#_Toc27433133)

[شناسه‌ها 5](#_Toc27433134)

[نماد های عملگرها 5](#_Toc27433135)

[کلمات کلیدی و رزرو شده 6](#_Toc27433136)

[توضیحات 6](#_Toc27433137)

[فضای تو خالی 6](#_Toc27433138)

[حائل و محصور کننده 6](#_Toc27433139)

[فرمت آزاد 6](#_Toc27433140)

[عبارت‌ها 6](#_Toc27433141)

[دستورات 6](#_Toc27433142)

[مراحل ترجمه ی زبان 6](#_Toc27433143)

[انواع داده‌های اولیه 6](#_Toc27433144)

[ساختمان داده‌ها 7](#_Toc27433145)

[شی گرایی 7](#_Toc27433146)

[کنترل ترتیب اجرا 8](#_Toc27433147)

# معرفی زبان go

زبان برنامه نویسی go یا golang یکی از زبان‌های جدید است که به سرعت به محبوبیت بالایی رسیده است. این زبان در سال 2007، در شرکت گوگل طراحی و ساخته شده است. این زبان که با هدف ارتقای و بهینگی در برنامه نویسی همروند و با تمرکز بر سیستم‌های مبتنی بر شبکه و پروژه‌های بیسار بزرگ و کدهای طولانی طراحی شده، در بسیاری جهات با زبان‌های دیگر متفاوت است. طراحان زبان Go به عمد بسیاری از قابلیت‌های متداول را از زبان حذف کرده‌اند. موارد زیر نمونه‌هایی از آن هستند.

* کلاس ها
* وراثت
* سازنده‌ها و مخرب‌ها (Constructors & Destructors)
* محاسبات شاره گر‌ها (Pointer Arithmetic)
* ارجاع this
* تمپلت‌ها (Templates)
* استثناها (Exceptions)
* سربار گذاری متدها (Method Overloading)
* سربار گذاری عملگر (Operator Overloading)
* و...

زبان Go استفاده‌های مختلف و زیادی دارد ولی بیشترین استفاده از آن برای طراحی و پیاده سازی میکرو سرویس‌های سمت سرور است. زبان Go یکی از کارامدترین و سریع‌ترین چهارچوب‌ها برای سرورهاست که قابلیت پردازش تعداد بسیار زیادی از درخواست‌ها را در زمان اندک دارد. Go زبانی است که برای نیازهای امروز برنامه نویسان طراحی شده است. تمامی سعی طراحان بر این بوده که بتوانند تناسبی بین قدرت، سرعت و سادگی ایجاد کنند. ایده‌هایی که در آن بکار رفته با دقت و وسواس زیادی انتخاب و پیاده سازی شده اند. طبق گفته تیم توسعه، بر سر بعضی از قابلیت‌های زبان تا یک سال بحث و تبادل نظر صورت گرفته است.

Rob Pike عنوان می‌کند که ایده اولیه آن‌ها ساخت یک زبان سیستمی بود که بتواند جایگاه مناسبی برای C++ باشد. در روزهای اول، حتی سایت پروژه هم Go را به عنوان زبانی برای برنامه نویسی سیستم معرفی می‌کرد. اما با گذشت زمان متوجه شدند که تاکید آن‌ها بر سادگی، باعث شده نظر برنامه نویسان زبان‌های سطح بالاتری مثل Python یا Ruby هم به Go جلب شود؛ این قضیه برای تیم توسعه هم جالب بود.

در واقع می‌توان گفت که Go با ظرافت فراوان قدرت و سرعت زبانی مثل C را با سهولت و سادگی زبانی مثل Python ترکیب کرده. به همین دلیل قادر است طیف بسیار وسیعی از برنامه‌ها را پوشش دهد، از برنامه‌های سیستمی گرفته تا برنامه‌های ساده چند خطی. چه کسی گفته که یک زبان سیستمی، حتماً باید سخت باشد؟

با اینکه Go مستقیما از طرف گوگل پشتیبانی می‌شود، ولی یک زبان کاملاً آزاد است و تحت کنترل هیچ شرکتی نیست Pike اعلام کرده که برای اثبات این موضوع، از قصد هیچ لوگو‌ای از شرکت گوگل را در سایت این زبان قرار نداده اند!

این یعنی:

سیاست کاری و تجاری هیچ شرکتی بر آینده زبان تاثیر نخواهد گذاشت.

توسعه و بهسازی زبان با سرعت بیشتری انجام می‌شود.

همه می‌توانند به طور شفاف تمام مراحل توسعه را دنبال کنند.

کتابخانه‌ها و فریم ورک‌های متن باز بیشتری برای زبان وجود خواهد داشت.

# تاریخچه زبان go

در سپتامبر سال 2007، Ken Thompson، Rob Pike دو تن از سرشناس ترین چهرها‌های دنیای نرم افزار، ایده ساخت یک زبان برنامه نویسی سیستمی جدید را با Robert Griesemer، یکی دیگر از مهندسین مشغول در Google، در میان گذاشتند. آن‌ها کار بر روی این زبان جدید را شروع کردند و در این مدت تعداد دیگری از همکارانشان همچون Russ Cox و Brad Fitzpatrick و Andrew Gerrand به تیم آن‌ها ملحق شدند. در نوامبر سال 2009، اولین نسخه آزمایشی زبان با پشتیبانی گوگل به صورت متن باز به عموم برنامه نویسان عرضه شد. از آن زمان تا کنون بیش از 300 نفر از برنامه نویسان داوطلب در توسعه این پروژه شرکت داشته اند.

از اولین ماه انتشار این زبان، شرکت‌های Startup زیادی استفاده از آن را شروع کردند. با اینکه زبان در نسخه آزمایشی به سر می‌برد، اما برای خیلی‌ها وجود نام Jeb Thompson و Rob Pike کافی بود تا از کیفیت زبان اطمینان حاصل کنند. شرکت‌های بزرگ تر اما، منتظر نسخه پایدار ماندند. تا اینکه در ماه مارس سال 2012، نسخه 1.0 از زبان برنامه نویسی Go به صورت پایدار منتشر شد.

در زیر لیست تعدادی از شرکت‌های استفاده کننده از این زبان را می‌بینید:

* Google
* YouTube
* Nokia Siemens
* Heroku
* CloudFlare
* Feedbooks
* SoundCloud

تقریباً در تمام اسامی بالا، از Go برای طراحی سیستم‌های Back-end استفاده شده است. جایی که زبان‌های اسکریپتی قدرت مناسب را ندارند و معمولا از JVM یا C# استفاده می‌شود؛ در خیلی از مواقع هم برای رسیدن به سرعت بالاتر، پای کدهای c و C++ به میان می‌آید. Go هنوز در اول راه است، و برای رسیدن به جایگاه زبان‌هایی مثل C++ و Java و Python و... راه درازی در پیش دارد، اما به نظر می‌رسد که برای شروع بسیار موفق عمل کرده است.

# کاربرد های زبان go

## برنامه نویسی میکرو سرویس‌ها و سرورها

یکی از بیشترین استفاده‌های زبان go مربوط به پیاده سازی و طراحی میکرو سرویس‌ها و سرورهای سریع است. زبان go به دلیل کامپایل native اجرای بسیار سریعی دارد و در جایی که سرعت بالا و پردازش‌های سنگین مد نظر باشد بهترین گزینه go است، چرا که نسبت به زبان c کد نویسی راحت‌تری دارد و هم نسبت به زبان‌هایی مثل php و python و حتی java و C# سرعت بیشتری دارد و سرعت اجرای آن با c قابل مقایسه است.

## برنامه نویسی سیستمی

یکی از معدود زبان‌های مدرنی که قابلیت کامپایل native دارد و اجرای آن نیاز به ماشین مجازی یا runtime خاصی ندارد، go است. به همین دلیل برای برنامه نویسی سیستمی می‌توان از go استفاده کرد. از قدیم بهترین زبان‌ها برای برنامه نویسی سطح پایین و طراحی برنامه‌های سیستمی، c و C++ بوده‌اند ولی با ظهور go این انحصار در حال شکسته شدن است.

## برنامه نویسی همروند

یکی از مهم‌ترین ویژگی‌های زبان go که در دل طراحی زبان قرار گرفته و زبان برای آن بهینه شده، قابلیت برنامه نویسی همروند و اجرای چند نخی و همزمان است. با توجه به توقف افزایش سرعت سخت افزارها به دلیل محدودیت‌های فیزیکی و چند هسته‌ای شدن CPU ها برای گرفتن کارایی بیشتر، مبحث برنامه نویسی همروند و اجرای موازی بیش از هر زمان اهمیت خود را نشان می‌دهد. به همین دلیل طراحان زبان go در زمان طراحی زبان این موضوع را مد نظر قرار داده و بهینه سازی‌ها و قابلیت‌های خوبی را برای این کار در زبان قرار داده‌اند. به همین دلیل یکی از بهترین زبان‌ها برای عملیات موازی و همروند زبان go است.

# مفسری یا کامپایلری؟

زبان go یکی از زبان‌های کامپایلری است. بر خلاف بسیاری از زبان‌های کامپایلری جدید مانند java و C# که کد منبع به کد میانی قابل تفسیر توسط ماشین مجازی و یا runtime تبدیل می‌شوند و در واقع عملیات کامپایل به طور کامل انجام نمی‌شود، زبان go مستقیماً به زبان ماشین ترجمه می‌شود. به همین دلیل می‌توان آن را یک زبان کامپایلری کامل یا واقعی نامید.

# ویژگی های مترجم

زبان golang دارای یک مترجم سطح پایین که کد‌ها را به زمان ماشین محلی تبدیل می‌کند است. این مترجم خود با زبان c توسعه داده شده و از سرعت و قدرت بالایی برخوردار است. همچنین مترجم زبان go قابلیت اجرای کد به صورت مفسری را نیز دارا می‌باشد که برای تست و خطایابی قابل استفاده است. مترجم زبان go برای بهینه‌سازی کاری انجام نمی‌دهد.

# انواع انقیادها و زمان های انقیاد

زبان go به دلیل کامپایلری بودن، از گروه زبان‌ها با انقیاد زودرس می‌باشد و بسیاری از انقیاد‌ها قبل از اجرا انجام می‌شوند.

## انقیادهای زمان طراحی زبان

شکل دستورات، انواع داده‌ها، انواع ساختمان داده‌ها، نمایش مقادیر عددی، نام‌های ممکن برای متغیرها و...

## انقیادهای زمان پیاده سازی

جزئیات مربوط به نمایش اعداد و عملیات های محاسباتی، نمایش دودویی اعداد، مجموعه مقادیر ممکن برای انواع داده‌ای اولیه و...

## انقیادهای زمان ترجمه

اسامی متغیرها، نوع متغیرها، انتساب به متغیرها، ساختار برنامه، محل نسبی اشیا در حافظه، چگونگی ذخیره کردن آرایه‌ها، توصیفگر آرایه، ساختار اشیا جدید و...

## انقیادهای زمان اجرا

انقیاد متغیرها به مقادیرشان، انقیاد متغیرها به محل حافظه

# عناصر نحوی زبان

## مجموعه‌ی کارکترها

زبان go از تمام مجموعه‌ی کارکتری UTF-8 پشتیبانی می‌کند. به عنوان مثال می‌توان برای نام متغیرها از حروف فارسی هم استفاده کرد!!

## شناسه‌ها

در زبان go شناسه‌ها باید با چیزی بجز اعداد و علائم رزرو شده شروع شود (مثلا حروف فارسی، چینی یا حتی اموجی) و می‌تواند با همان علائم به علاوه‌ی اعداد ادامه یابد.

## نماد های عملگرها

+ & += &= && == != ( ) - | -= |= || < <= [ ] \* ^ \*= ^= <- > >= { } / << /= <<= ++ = := , ; % >> %= >>= -- ! ... . : &^ &^=

+ جمع integers, floats, complex values, strings

- تفریق integers, floats, complex values

\* ضرب integers, floats, complex values

/ تقسیم integers, floats, complex values

% باقیمانده integers

& بیتی AND integers

| بیتی OR integers

^ بیتی XOR integers

&^ صفر کردن بیت‌ها integers

<< شیفت به چپ integer << unsigned integer

>> شیفت به راست integer >> unsigned integer

== برابری

!= نابرابری

< کوچک‌تر

<= کوچک‌تر مساوی

> بزرگ‌تر

>= بزرگ‌تر مساوی

&& منطقی AND p && q یعنی "if p then q else false"

|| منطقی OR p || q یعنی "if p then true else q"

! منطقی NOT !p یعنی "not p"

& آدرس

\* مقدار اشاره‌گر | تعریف اشاره‌گر

<- دریافت

## کلمات کلیدی و رزرو شده

break default func interface select case defer go map struct chan else goto package switch const fallthrough if range type continue for import return var

## توضیحات

در زبان go کامنت‌ها با /\* شروع شده و با \*/ پایان می‌یابند. همچنین کامنت یک خطی // هم قابل استفاده است.

## فضای تو خالی

در زبان go فضاهای خالی ارزشی ندارند اما خود go استاندارد سخت گیرانه‌ای نسبت به مرتب بودن کد دارد و در مواردی اگر کد با شکل استاندارد نباشد کد کامپایل نخواهد شد. البته خود کامپایلر go قابلیت مرتب کردن کد را نیز دارا می‌باشد.

## حائل و محصور کننده

در زبان go دستورات می‌توانند با پایان خط یا ; به پایان برسند. کامپایلر go در زمان مرتب سازی کد به خواست برنامه نویس، خودش ; را حذف و ادامه‌ی کد را به خط بعد می‌برد.

در زبان go از {} به عنوان محصور کننده‌ی قطعات کد استفاده می‌شود.

## فرمت آزاد

زبان go هم فرمت آزاد دارد و هم ندارد. به این صورت که کامپایلر در صورت مرتب کردن کد، خودش فاصله گذاری‌ها را تصحیح می‌کند و نحوه‌ی فاصه گذاری‌ها کاملاً معین و سخت‌گیرانه هستند.

## عبارت‌ها

## دستورات

دستورات در زبان go ساخت یافته می‌باشد، به دین صورت که زبان دارای توابع و از ساختار دستوری تو در تو پشتیبانی می‌کند.

# مراحل ترجمه ی زبان

# انواع داده‌های اولیه

زبان go از اکثر انواع داده‌های اولیه‌ی استاندارد پشتیبانی می‌کند که بر اساس استاندارد IEEE پیاده سازی و ذخیره می‌شوند. انواع داده‌ای زبان go شامل موارد زیر است:

bool نوع داده منطقی است که می‌تواند شامل مقادیر true یا false باشد و 8 بیت فضا در حافظه اشغال می‌کند.

uint8 برای ذخیره سازی داده‌ی عددی بدون علامت است که 8 بیت فضا اشغال می‌کند. نوع byte نام دیگری برای همین نوع است.

uint16 برای ذخیره سازی داده‌ی عددی بدون علامت است که 16 بیت فضا اشغال می‌کند.

uint32 برای ذخیره سازی داده‌ی عددی بدون علامت است که 32 بیت فضا اشغال می‌کند.

uint64 برای ذخیره سازی داده‌ی عددی بدون علامت است که 64 بیت فضا اشغال می‌کند.

int8 برای ذخیره سازی داده‌ی عددی علامت دار است که 8 بیت فضا اشغال می‌کند.

int16 برای ذخیره سازی داده‌ی عددی علامت دار است که 16 بیت فضا اشغال می‌کند.

int32 برای ذخیره سازی داده‌ی عددی علامت دار است که 32 بیت فضا اشغال می‌کند. نوع rune نام دیگری برای این نوع است که از نظر مفهومی برای ذخیره‌ی کاراکترها استفاده می‌شود.

int64 برای ذخیره سازی داده‌ی عددی علامت دار است که 64 بیت فضا اشغال می‌کند.

float32 برای ذخیره سازی داده‌های عددی اعشاری است که 32 بیت فضا اشغال می‌کند.

float64 برای ذخیره سازی داده‌های عددی اعشاری است که 64 بیت فضا اشغال می‌کند.

complex64 برای ذخیره سازی داده‌های عددی مختلط است که 64 بیت فضا اشغال می‌کند.

complex128 برای ذخیره سازی داده‌های عددی مختلط است که 128 بیت فضا اشغال می‌کند.

int این نوع عددی، به طور کلی برای ذخیره سازی اعداد علامت دار استفاده می‌شود، کامپایلر اندازه‌ی این نوع متغیر را خودکار و به صورت بهینه انتخاب می‌کند.

uint این نوع عددی، به طور کلی برای ذخیره سازی اعداد بدون علامت استفاده می‌شود، کامپایلر اندازه‌ی این نوع متغیر را خودکار و به صورت بهینه انتخاب می‌کند.

uintptr نوعی با قابلیت ذخیره سازی مقادیر اشاره‌گرهاست. اندازه‌ی این متغیر با توجه به معماری سیستم مشخص می‌شود.

string برای ذخیره سازی رشته‌ها (متن) استفاده می‌شود.

Pointer زبان go از انواع اشاره‌گر به صورت کامل پشتیبانی می‌کند. برای تعریف یک اشاره‌گر کافی است یک کاکتر \* قبل از نوع مورد نظر استفاده کرد.

# ساختمان داده‌ها

زبان go برای سادگی تنها از سه نوع ساختار داده‌ای زیر استفاده می‌کند. این سه نوع از انواع پر استفاده و استاندارد هستند که دراکثر زبان‌های برنامه نویسی با نام‌های مختلف وجود دارند.

Array: زبان go از ساتار داده‌ای آرایه پشتیبانی کاملی دارد.

Slice: این نوع داده‌ای برای در اصل نماینده‌ی زیر مجموعه‌ای از یک آرایه است.

Map: این نوع داده‌ای مجموعه‌ای از کلیدها و مقادیر مرتبط با آن کلید است. نوع کلید و مقدار می‌تواند هر چیزی باشد ولی قابل تغییر نیست.

# شی گرایی

زبان Go به عمد از شی گرایی پشتیبانی نمی‌کند. دلیل این کار رویکرد سیستمی زبان Go و کم کردن پیچیدگی‌های زبان است. البته این زبان از Struct ها و Interface ها پشتیبانی می‌کند تا بتواند نیازهای ضروری را براورده کند.

# کنترل ترتیب اجرا

کنترل اجرا در زبان go شاید متفاوت‌ترین بخش زبان گو باشد چرا که این زبان از ابتدا بر اساس اجرای همروند ساخته شده و امکانات مختلف و زیادی برای بهره برداری از این امکانات در زبان گنجانده شده.

کنترل اجرا در زبان go در توابعی که شامل همروندی نیستند، مانند یک زبان ساخت یافته‌ی استاندارد است. هر تابعی که به صورت همروند فراخوانده شود، همزمان با تابع فراخواننده و توابع دیگر فراخوانی شده به صورت همروند اجرا می‌شود. همچنین پیشروی اجرا در زمان دریافت و ارسال داده از کانال‌ها متوقف می‌شود.