

# 1. GoKYCH Typst

Typst is a GoKYCH 5 (md / wikidot / html / bbcode / typst), which is a markup language for generating HTML or PDF documents. It is a simple, declarative language that is easy to learn and use. Typst is a Go-based language that is designed to be a simple, declarative language for generating HTML or PDF documents. It is a simple, declarative language that is easy to learn and use. Typst is a Go-based language that is designed to be a simple, declarative language for generating HTML or PDF documents. It is a simple, declarative language that is easy to learn and use.

## 1.1. Basic Syntax

```
#import "template.typ": template

#template[
  =
  ...
]
```

#import "template.typ" template; #template[ ... ] ... template[] ... —

## 1.2. Text

```
= (H1)
== (H2)
=== (H3)
```

1-3 template.typ; typst

## 1.3. Lists

- \* / \_ / #underline
- : typst mark
- : #link("https://example.com")

## 1.4. Lists

:

- 
- 
- 

:

- 
- 
- 

## 1.5. Lists

typst



## 1.10. `table`

```
#table(
  columns: (auto, 1fr, auto),
  align: (left, left, right),
  [1], [2], [3],
  [A], [B], [1],
  [C], [D], [2],
)
```

## 1.11. `typst`

template.typ `typst` H1 `typst` 1.1 / 1.1.1 `typst` (`PDF`) `typst` `nonumber`:

```
= typst <nope>
```

## 1.12. `typst`

`typst` `typst`,Go `typst`:

- **create** `typst`:`typst`(`typst`),`typst` 400,slug `typst`, `typst` `typst` CLI `typst`
- **update** `typst`:`typst` title / content `typst`,`typst` 400, `typst` – `typst`

`typst`:

```
cd /opt/gokych/data/typst
typst compile your-article.typ your-article.html
```

`typst`:

1. `typst` #template[ ... ] `typst`
2. #import `typst`(typo `typst`)
3. #image `typst`
4. typst `typst`(`typst` / `typst` / `typst` / `typst`)

## 1.13. `typst` template

Go `typst` template.typ `typst`,`typst` `typst`

```
workspace typst typst /opt/gokych/data/typst/ ,typst cfg.App.DataDir + "/typst" typst (typst cmd/gokych/main.go typst
typst.SetWorkspaceDir typst)
```

## 1.14. `typst`

`typst`,`typst`,`typst` – `typst` `typst`typst `typst` 1 `typst` semaphore slot, `typst` 4 `typst`,30s  
timeout(`typst` typst `typst`)