What should I know about GIT?

- Source control
- Branching
- Merge
- Rebase
- Cherry pick
- Stash
- Tag
- Etc.


TS

Why I should use Typescript?

- Less bugs
- Development performance
- Features: type checking, autocompletion, source documentation and etc.
- Responsive support from the Typescript team
- Starts and ends with JavaScript
- Beautiful and elegant
- Etc.

Getting started

- https://nodejs.org/en/
- npm install -g typescript (or install locally in dev dependencies)
- tsc helloworld.ts

How to compile

tsc helloworld.ts

tsconfig.json

Settings

- target
- removeComments
- outDir
- sourceMap
- outFile
- files
- Etc.

Variables

- Use ES6 **let** and **const** to declare variables in Typescript.

- let user: User = new User();
- const cardNumber: string = “**** **** **** 1111”;

Types

- boolean
- number
- String
- Object
- Array
- Tuple
- enum
- null
- undefined
- void
- never
- any

Type declaration

- declare type primitive = number | string | boolean| null | undefined;

Type conversion

- <type>variable;
- as

What should I know about **classes** in TS?

- fields
- methods
- constructor
- static fields and methods
- private/public/protected
- get/set
- readonly

Inheritance and abstract classes

- basic inheritance mechanism in general the same as in ES6

- abstract class can’t be instantiated
- but more concrete implementation can be
- example: abstract Figure, concrete Square and Circle.

What should I know about **interfaces** in TS?

- syntax
- optional and readonly properties
- implementation
- Interface inheritance
- Function and array interfaces

Generics

- Use generics when something should work with any data types.
- You can use them with functions, classes, interfaces.

- Example: promisify, Map class.

Namespaces

- Namespaces contains a group of classes, functions, interfaces, variables, other namespaces, etc.
- Use **namespace** to declare namespaces.
- Use **exports** for using entities in other namespaces.

Modules

TS support the following modules:

- AMD (Asynchronys Module Defenition)
- CommonJS
- UMD (Universal Module Defenition)
- System
- ES 2015

- Use **export/import** features.

- Example: User and UserService

Decorators

- Allow to add metadata to classes or their members for changing their behavior without changing their code.
- Decorator – function, which can apply on class, field, methods, getter, setter, parameters, etc.
- Set **experimentalDecorators: true** for using decorators**.**

Resources

- https://git-scm.com/

- https://www.typescriptlang.org/
- https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html
- https://metanit.com/web/typescript/1.1.php
- https://habr.com/ru/post/471026/