

* Function's in TS:-

1) Function Declaration:-

```
function add(a: number, b: number): number {  
    return a + b;  
}
```

Annotations:
- `add`: fun Name
- `a: number, b: number`: parameter type
- `number`: return type

2) Function expression

```
const multiply = function(x: number, y: number): number {  
    return x * y;  
}
```

3) Arrow Function

```
const divide = (x: number, y: number): number => {  
    return x / y;  
};
```

4) let greet: (name: string) => string;

```
greet = function(name) {  
    // ...  
}
```

5.) Optional Parameters: (---?)

```
function logMessage (message: string,  
    userId?: number): void {  
    console.log (userId ? '[${userId}] ${message}'  
        : message);  
}
```

6.) Default Parameters

```
function power (base: number, exponent: num  
    - base = 2): number {  
    return base ** exponent;  
}
```

7.) Rest Parameters

```
function sumAll (...nums: number[]): number {  
    return nums.reduce((a, b) => a + b, 0);  
}
```

8.) Function Overloading:

```
function combine (a: string, b: string): string;  
function combine (a: number, b: number): number;  
- or;  
function combine (a: any, b: any) {  
    return a + b;  
}
```


9) Methods inside Objects/classes

```
const calculator = {  
  add(x: number, y: number): number {  
    return x + y;  
  }  
};
```

```
class Person {  
  constructor(public name: string) {}  
  
  greet(): void {  
    console.log(`Hi, I'm ${this.name}`);  
  }  
}
```

10) Union Types (1)

```
function findUserAge(name: string):  
  name | null {
```

```
  if (name === 'Anant') {  
    return 19;  
  } else {  
    return null;  
  }  
}
```

Note:- When calling, we have to handle both types.

11. Unknown:

It could be anything, but you must check before using it.

12. Generics

function identity ^{↙ is a type} <T> (value: T): T {
 return value;
}

13. Void and Never ~~infinite loop/error~~

```
function log(msg: string): void {  
    console.log(msg);  
}
```

```
function fail(msg: string): never {  
    throw new Error(msg);  
}
```