



BUILT-IN UTILITY TYPES



RECORD

با استفاده از یوتیلیتی تایپ رکورد میتونید تایپ آبجکت خوانا و هدفمند تری داشته باشید.



```
export type Product = 'Pant' | 'Shirt' | 'Socks';

export type ProductCount = {
  | key: Product]: number
}
```

حتی در این روش قدیمی نمیتوانید از یونیون تایپ ها برای مقدار key ها استفاده کنید.



```
export type Product = 'Pant' | 'Shirt' | 'Socks';

type ProductCount = Record<Product , number>;
```

حتما نیاز به ساخت یک تایپ جدید هم نیست میتوانید آنرا را مستقیما به آبجکت مورد نظر تخصیص دهید



OMIT

با استفاده از یوتیلیتی تایپ امیت میتونید تایپ جدیدی بسازید که بعضی از پراپرتی های تایپ هدف رو نداشته باشه.



```
type UserInformation = {  
  phoneNumber: number,  
  address: string[],  
  nationalCode: number,  
  firstName: string,  
  lastName: string  
}
```



```
type UserPrivateFields = {  
  nationalCode: number,  
  phoneNumber: number,  
  address: string[],  
}
```



```
type UserInformation = {  
  phoneNumber: number,  
  address: string[],  
  nationalCode: number,  
  firstName: string,  
  lastName: string  
}
```

نسبت به کد بالا هم خوانا تره هم کوتاه تر

میتونید تایپ جدید هم درست نکنید و همینو مستقیم بدید به آبجکتتون

```
type UserPrivateFields = Omit<UserInformation , 'firstName' | 'lastName'>;
```



PICK

با استفاده از یوتیلیتی تایپ پیک می‌توانید تایپ جدیدی بسازید که فقط بعضی از پراپرتی‌های هدف رو داشته باشه (دقیقا بر عکس Omit).



```
type UserInformation = {  
  phoneNumber:number,  
  address:string[],  
  nationalCode:number,  
  firstName:string,  
  lastName:string  
}
```

```
type UserPublicInformation = {  
  firstName:string,  
  lastName:string  
}
```



```
type UserInformation = {  
  phoneNumber:number,  
  address:string[],  
  nationalCode:number,  
  firstName:string,  
  lastName:string  
}
```

نسبت به کد بالا هم خوانا تره هم کوتاه تر

می‌تونید تایپ جدید هم درست نکنید و همینو
مستقیم بدید به آبجکتتون

```
type UserPublicInformation = Pick<UserInformation, 'firstName'|'lastName'>;
```



READONLY

با استفاده از یوتیلیتی تایپ ریدآنلی میتونید تایپ جدیدی بسازید که همه پراپرتی های آن فقط قابلیت خوانده شدن داشته باشند.



```
type UserInformation = {  
  nationalCode: number;  
  firstName: string;  
  lastName: string;  
};
```

```
type ReadonlyUserInformation = {  
  readonly nationalCode: number;  
  readonly firstName: string;  
  readonly lastName: string;  
};
```



```
type UserInformation = {  
  nationalCode: number;  
  firstName: string;  
  lastName: string;  
};
```

```
let readOnlyUserInformation : Readonly<UserInformation>;
```

PARTIAL

با استفاده از یوتیلیتی تایپ پارشیال میتونید تایپ جدیدی بسازید که پراپرتی های آن آپشنال شوند.



```
type UserInformation = {  
  nationalCode: number;  
  firstName: string;  
  lastName: string;  
};
```

```
type OptionalUserInformation = {  
  nationalCode?: number;  
  firstName?: string;  
  lastName?: string;  
};
```



```
type UserInformation = {  
  nationalCode: number;  
  firstName: string;  
  lastName: string;  
};
```

```
let optionalUserInformation : Partial<UserInformation>;
```

← خلاصه و بدون کار اضافه