

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    //enum lista
    enum weekday {Monday, Tuesday, Wednesday, Thursday, Friday, Saturday,
Sunday};
    enum weekday today = Monday; //inicializálás Literálás beállítjuk a Today
értékét

    switch (today)
    {
        case Sunday:
            printf("Today is Sunday.");
            break;
        case Monday:
            printf("Today is Monday.");
            break;
        case Saturday:
            printf("Today is Saturday.");
            break;
        default:
            printf("Whatever.");
            break;
    }
    return 0;
}

```

Mit csinál a kód röviden?

- Létrehozol egy `enum weekday` típust a hét napjaival.
- Beállítod, hogy `today = Monday`.
- A `switch` szerkezet segítségével eldöntöd, hogy milyen nap van, és kiírod az adott nap nevét, ha az Sunday, Monday vagy Saturday.
- Más esetben a `default` ág fut le, és kiírja, hogy „Whatever.”

Néhány plusz javaslat, ha tovább szeretnéd fejleszteni vagy olvashatóbbá tenni:

1. Enum elemek értéke alaptól 0-tól kezdődik, így:

- Monday = 0
- Tuesday = 1
- ...
- Sunday = 6

2. Érdekes lehet az enumot typedef-el kombinálni, hogy egyszerűbben használd:

```
typedef enum {  
    Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday  
} weekday;  
  
weekday today = Monday;
```

1. Ha a hét minden napját szeretnéd kezelni a switch-ben:

```
switch (today) {  
    case Monday: printf("Today is Monday.\n"); break;  
    case Tuesday: printf("Today is Tuesday.\n"); break;  
    case Wednesday: printf("Today is Wednesday.\n"); break;  
    case Thursday: printf("Today is Thursday.\n"); break;  
    case Friday: printf("Today is Friday.\n"); break;  
    case Saturday: printf("Today is Saturday.\n"); break;  
    case Sunday: printf("Today is Sunday.\n"); break;  
    default: printf("Invalid day.\n"); break;  
}
```

1. Ha szeretnéd, beolvashatod a napot is felhasználói bemenetként, és annak megfelelően állítod be az értéket.
