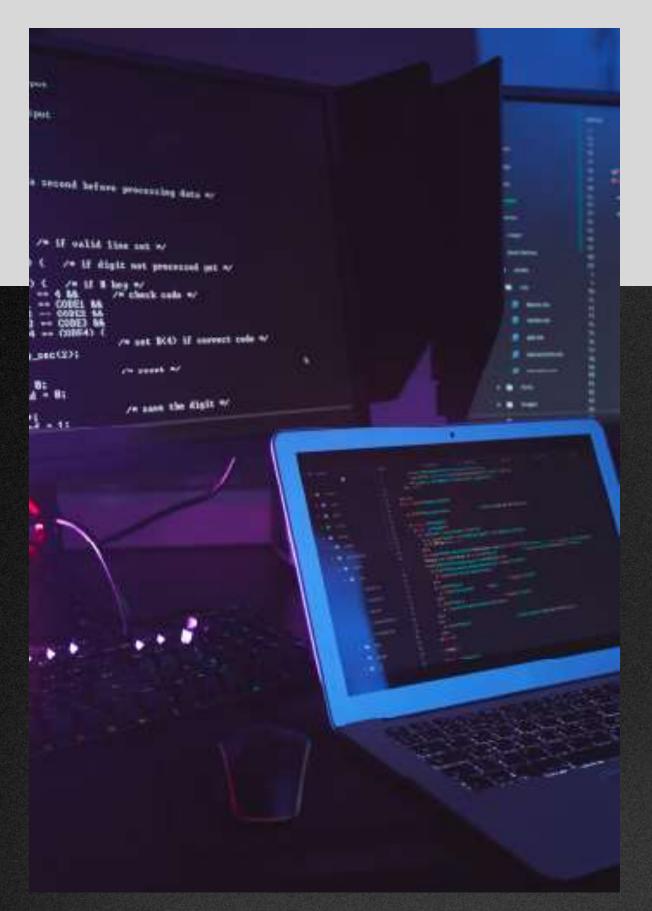
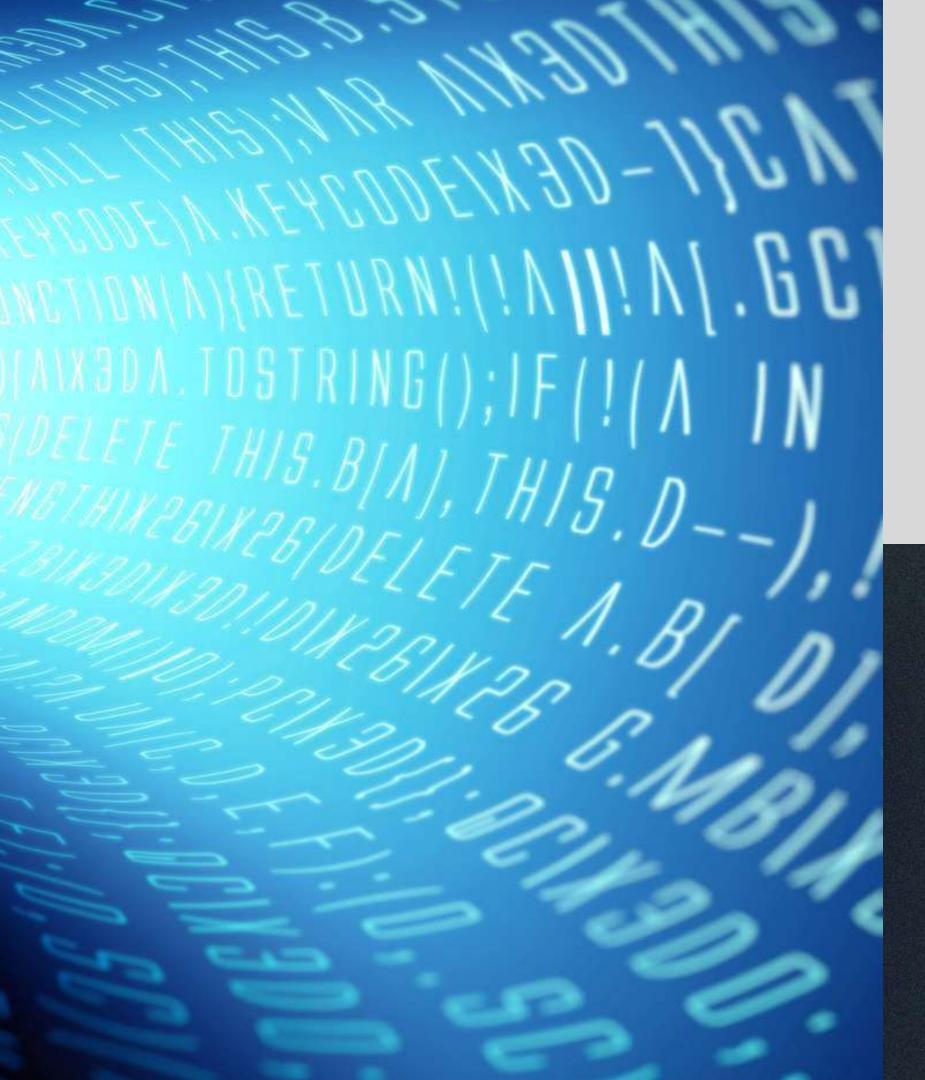
```
02
          def on key_press(self, symbol, modifiers):
83
84
               if self.context_index == -1:
85
                    if symbol == key.UP and not self.active_index == 0:
86
                        self.menu_labels[self.active_index].color = [255, 255, 255, 255]
87
88
                        self.mags dt = self.get act color mag()
89
                    elif symbol == key.DOWN and not self.active in
90
                         self.menu_labels[self.active_index].color = ...
91
                         self.active index += 1
92
                         self.mags dt = self.get act color mag()
93
                    elif symbol == key.ENTER:
94
                         if self.active index == 3:
95
                             pyglet.app.exit()
96
                         else:
97
                             self.context index = self.active index
98
                    elif symbol == key.ESCAPE:
                         if self.context index == -1:
99
                             pyglet.app.exit()
100
                                                                        ŠTĚPÁN HÁVA 4ITA
                         else:
101
                             self.context_index = -1
102
               elif self.context index == 1:
103
                    if symbol == key.ESCAPE:
104
                        self.context_index = -1
105
                        self.cur_game.on key press(symbol, modifiers)
                    else:
106
107
                    if symbol == key.ESCAPE:
                else:
108
                        self.context index = -1
109
```



## GOUSESSIONS?

- Mechanismus pro uchování dat uživatele během práce s webovou aplikací
- Data se ukládají na serveru a jsou dostupná napříč požadavky uživatele
- Každá session má unikátní identifikátor (Session ID), který ji propojuje s klientem



# SESSIONS V ASP.NET

- V ASP.NET se sessions spravují pomocí HttpContext.Session
- Data lze ukládat jako klíč-hodnota
- Pro práci se sessions je nutné nejprve zapnout session storage

```
HttpContext.Session.SetString("user", "Petr");
string user = HttpContext.Session.GetString("user");
```

#### ČTENÍ A SPRÁVA SESSIONS

- Čtení dat ze session
  - Data Ize kdykoliv přečíst během životnosti session

```
if (HttpContext.Session.GetString("user") != null)
{
    string user = HttpContext.Session.GetString("user");
}
```

- Mazání dat:
  - Smazání konkrétní hodnoty
     HttpContext.Session.Remove("user");
  - <u>Ukončení celé session</u>
     <u>HttpContext.Session.Clear();</u>

#### MOZNOSTI UKLADANI SESSION DAT

- 1. In-Process data jsou uložena v paměti serveru (nejrychlejší)
- 2. State Server data jsou uchovávána v samostatné službě (lepší škálovatelnost)
- 3. SQL Server session data jsou uložena v databázi (trvalé uložení)
- 1. sessionState mode="InProc"
- 2. sessionState mode="StateServer"
- 3. sessionState mode="SQLServer"

### SISSINS IS. FINITES

| Sessions  | Cookies                           |
|---|-----------------------------------|
| Data jsou na serveru                            | Data jsou uložena u klienta       |
| Bezpečnější                                     | Méně bezpečné (může být ukradeno) |
| Kratší životnost (zánik při zavření prohlížeče) | Může být platné dlouhou dobu      |
| Nepotřebuje podporu od klienta                  | Musí být povolené cookies         |



### POUŽITÍ SESSIONS

- Zapamatování preferencí uživatelů, např. preferovaná měna na stránce, preferovaný jazyk
- Zapamatování košíku na různých e-shopech
- Uchování přihlášení
- Přenášení dat bez použití cookies

