

# Szybka powtórka

- IP
- DNS
- TCP
- Klient i Serwer
- HTTP
- HTML
- JSON

# CSS - Cascading Style Sheets



# CSS - rules

The diagram illustrates the components of a CSS rule. It shows a selector 'p' followed by a block of declarations enclosed in curly braces. Labels with arrows identify each part: 'selector' points to 'p', 'declaration' points to the entire block, 'property' points to 'color', and 'value' points to 'red'.

```
selector  →  p {  
              color: red;  
              font-size: 24px;  
            }  
              ↑      ↑  
            property value  
              ← declaration
```

# CSS - selectors

- id selector (#mainTitle)
  - `<h1 id="mainTitle">...</h1>`
- class selector (.large)
  - `<p class="large">...</p>`
- attribute selector (input[type="text"])
  - `<input type="text">`
- element selector (p)
  - `<p>...</p>`
- universal selector (\*)

# CSS - selectors

```
p {  
  color:  red;  
}
```

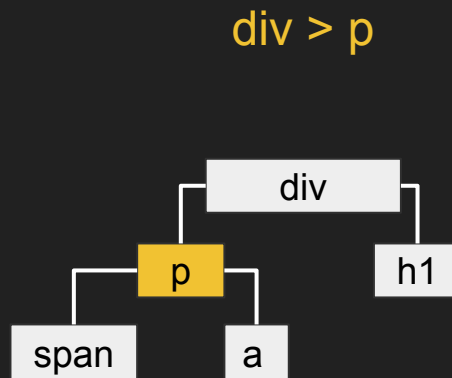
```
p, .large {  
  font-size: 24px;  
}
```

```
p, h1 {  
  color:  red;  
}
```

```
.large.is-hidden {  
  font-size: 24px;  
  display: none;  
}
```

# CSS - combinators

- descendant (**space**)
- direct child (**>**)
- adjacent sibling (**+**)
- general sibling selector (**~**)
- or (**,**)
- and (**div.large**)



# CSS - Kaskadowość



# CSS - Inline styles

```
<p id="green" style="color:  gold;">...</p>
```



# CSS - specificity

Klasyfikacja na podstawie rodzaju selektora

- ID selectors - 1-0-0
- class selector - 0-1-0
- type selector - 0-0-1

Wartość selektora p = 0-0-1

Wartość selektora p.blue = 0-1-1

```
p {  
  color:  red;  
}  
  
p.blue {  
  color:  blue;  
}
```

# CSS - order of appearance i !important

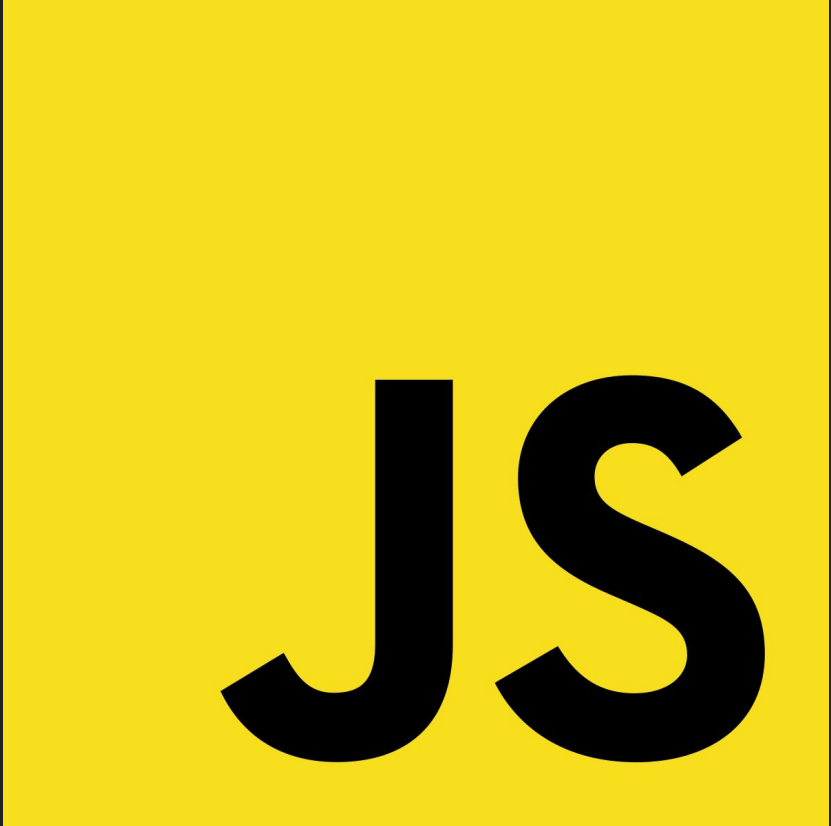
```
p {  
  color:  red;  
}
```

```
p {  
  color:  purple;  
}
```

```
p {  
  color:  red !important;  
}
```

```
p {  
  color:  purple;  
}
```

JS - JavaScript

The JavaScript logo, consisting of the letters 'JS' in a bold, black, sans-serif font, centered within a bright yellow square. The square is positioned in the lower right portion of the dark gray background.

**JS**

# Eventy DOM

Najczęstsze eventy:

- **click** (kliknięcie myszy)
- **mousemove** (najechnięcie myszą na element)
- **change** (użytkownik korzysta z inputów)
- **submit** (zatwierdzany jest formularz)
- **keydown** (naciśnięcie przycisku na klawiaturze)

# Pobieranie elementów

Pojedyncze elementy:

- `document.getElementById("intro")`
- `document.querySelector("p.red")`

Lista elementów:

- `document.getElementsByTagName("p")`
- `document.getElementsByClassName("large")`
- `document.querySelectorAll("p.red")`

# Cache

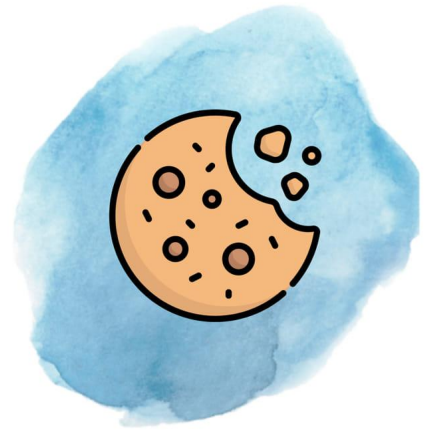


# Cookies and Storage

	Cookies	Local Storage	Session Storage
Pojemność	4kb	10mb	5mb
Standard	HTML4/HTML5	HTML5	HTML5
Dostępne z	Każdego okna	Każdego okna	Tej samej zakładki
Czas wygaśnięcia	Ustawiane manualnie	Nigdy	Zamknięcie zakładki
Miejsce przechowywania	Przeglądarka i serwer	Przeglądarka	Przeglądarka
Wysyłane z żądaniem	Tak	Nie	Nie

# Cookies

**Set-Cookie:** key=value; Secure; HttpOnly;





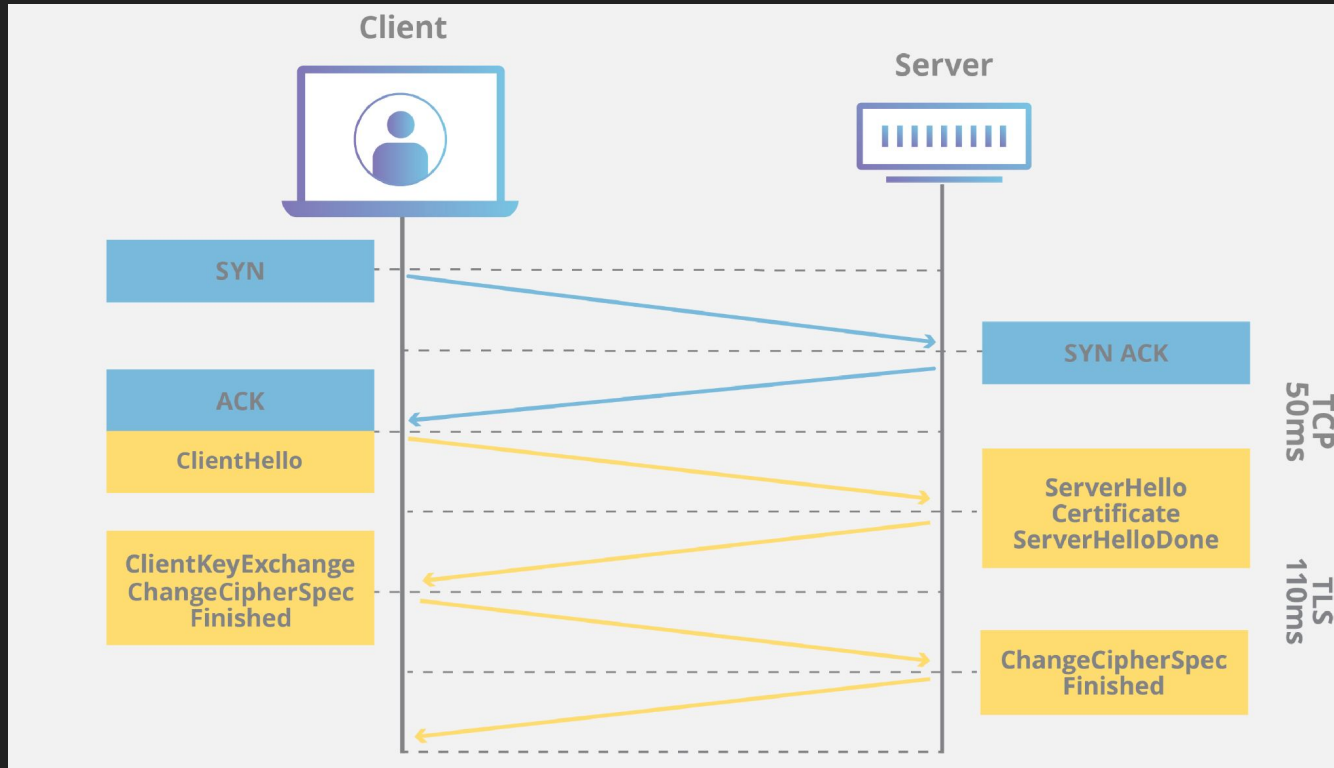
## Różne formy ataków:

- XSS - Cross Site Scripting
- CSRF (XSRF) - Cross Site Request Forgery
- SQL Injection
- Man in the middle
- Spoofing

# HTTPS - HyperText Transfer Protocol **Secure**



# SSL/TLS

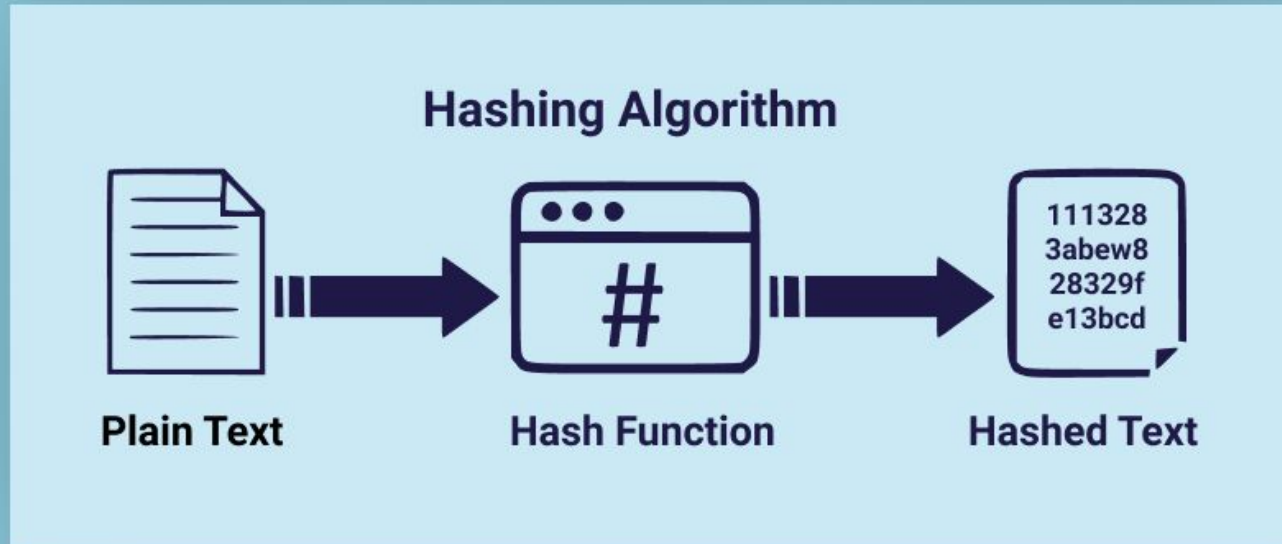


# Kryptografia


- Hash
- Salt
- Hmac
- Szyfrowanie symetryczne
- Szyfrowanie asymetryczne

<https://haveibeenpwned.com/Passwords>

# Hash

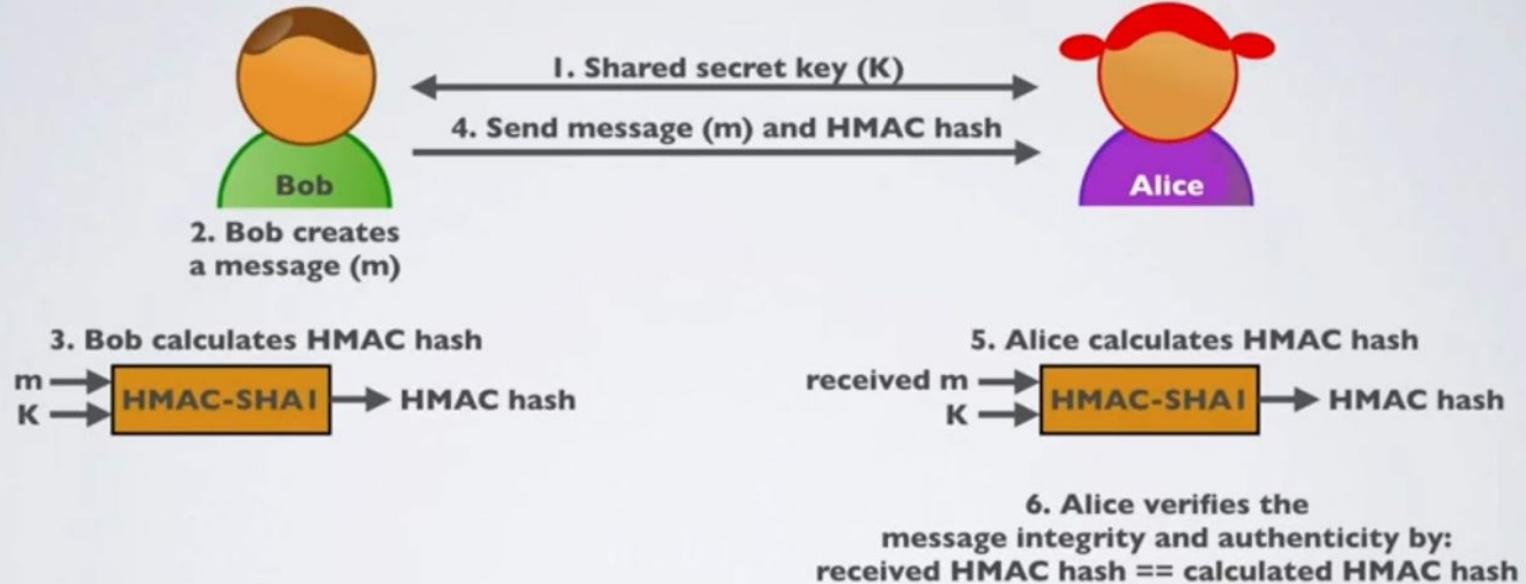


# Salt

				
Password	p4s5w3rdz	p4s5w3rdz	p4s5w3rdz	p4s5w3rdz
Salt	-	-	et52ed	ye5sf8
Hash	f4c31aa	f4c31aa	1vn49sa	z32i6t0

# HMAC - Hash-Based Message Authentication Code

## HOW HMAC IS USED

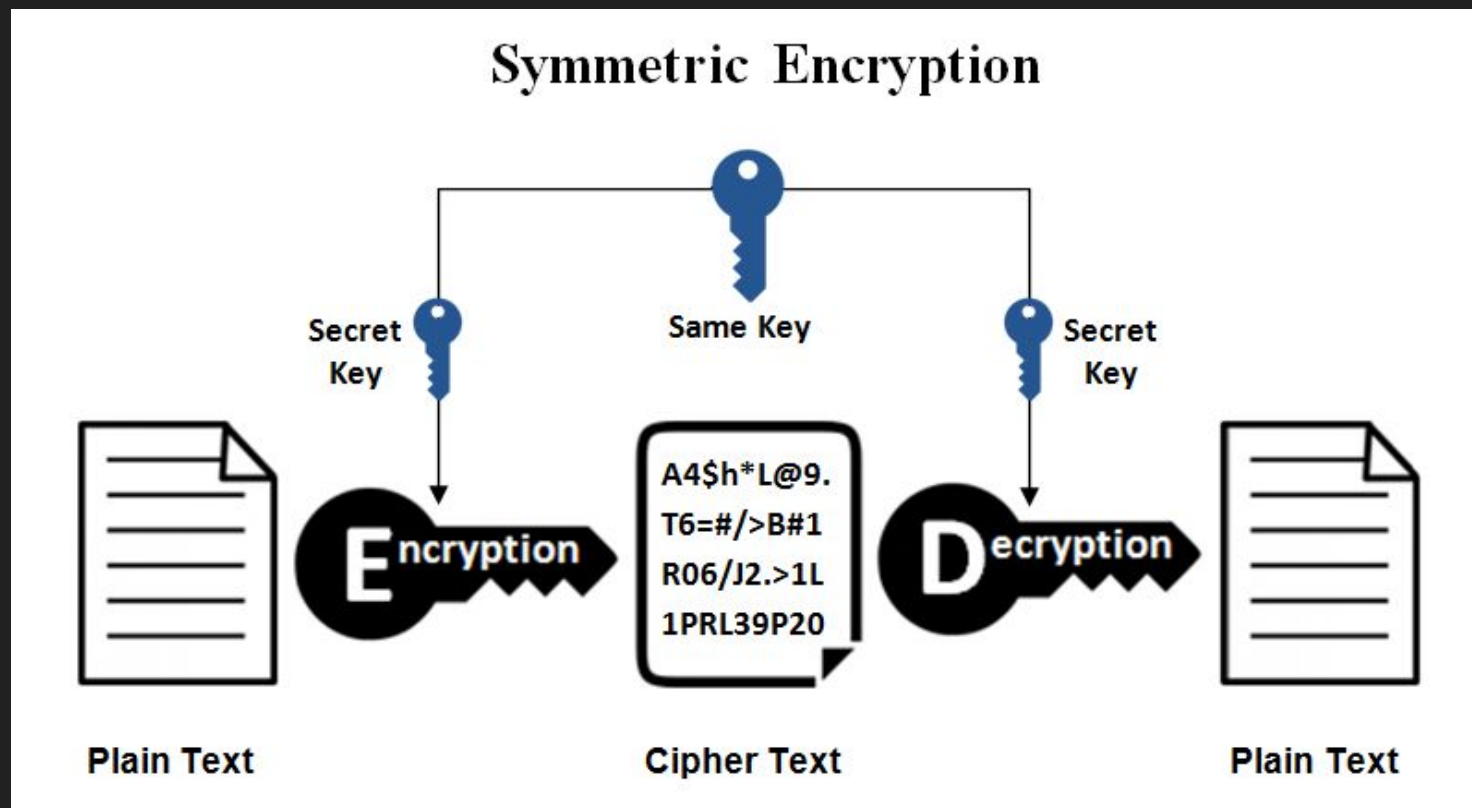


# Encryption - szyfrowanie

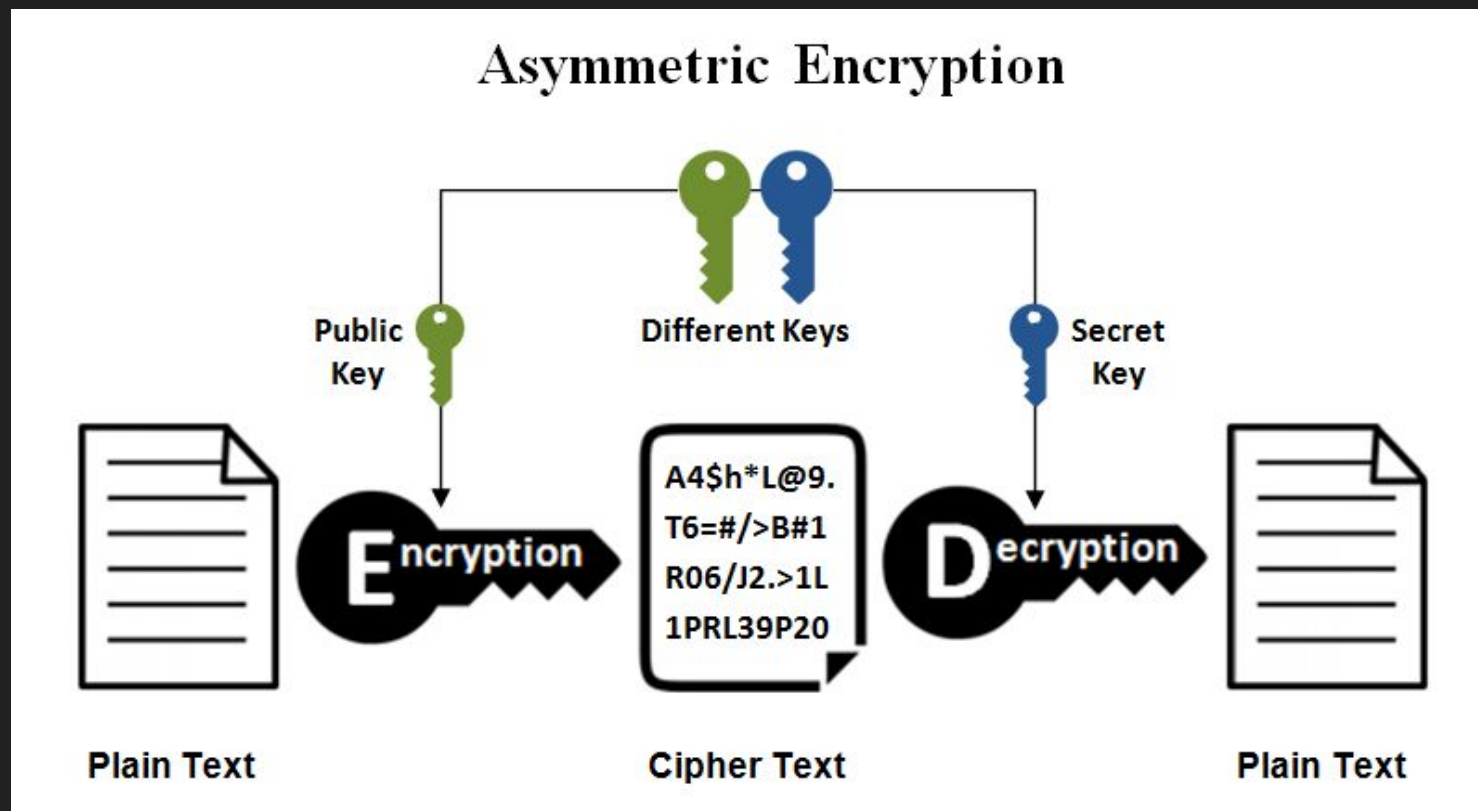
- symetryczne (symmetric):
  - AES-256
- asymetryczne (asymmetric):
  - RSA



# Szyfrowanie symetryczne



# Szyfrowanie asymetryczne



# JWT - JSON Web Token

