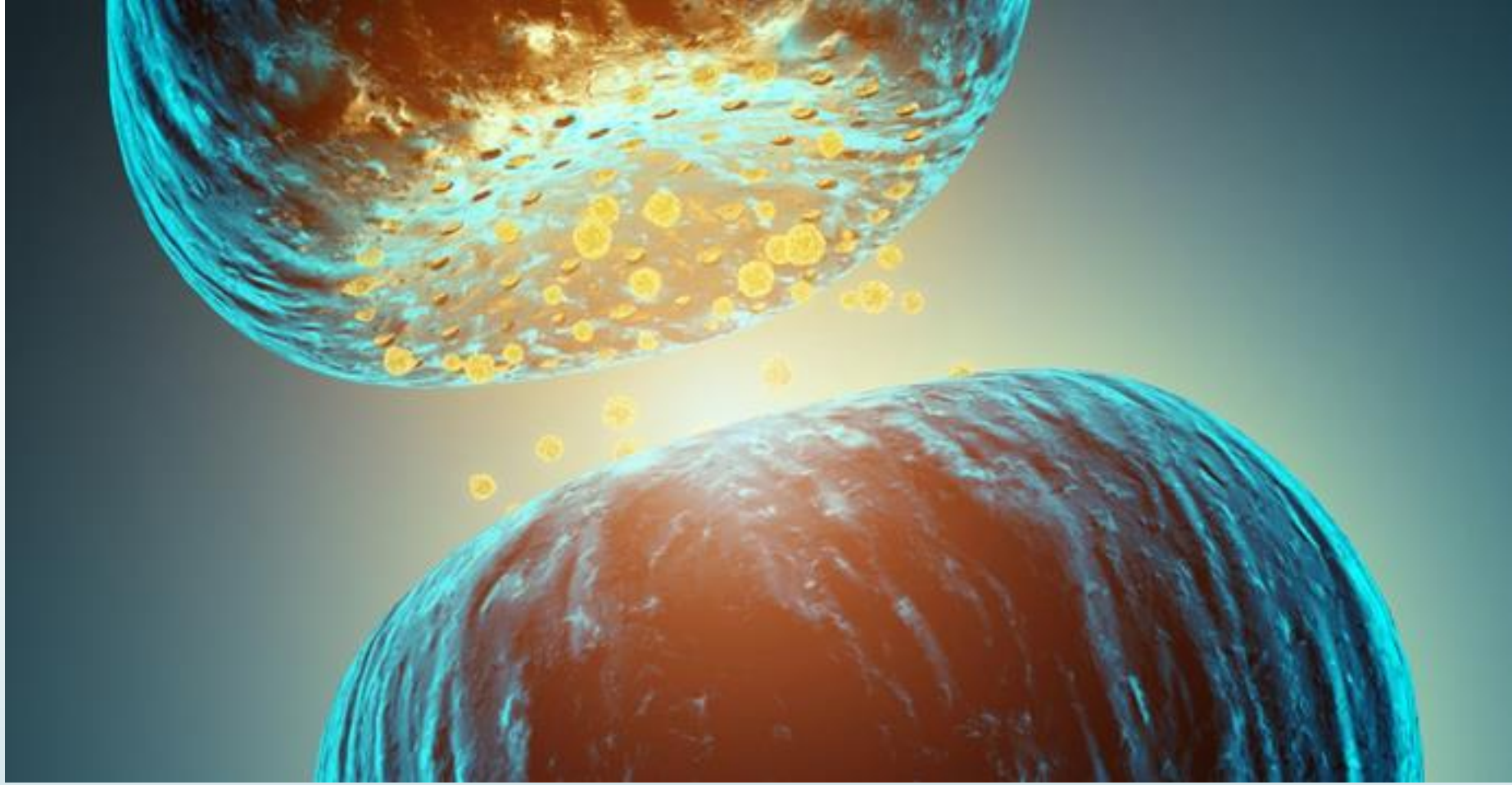
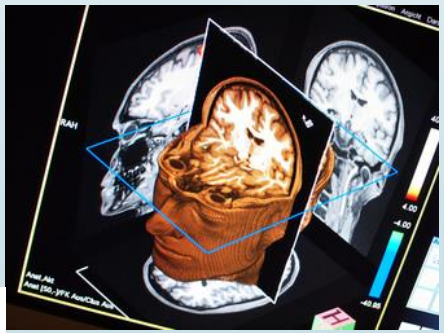


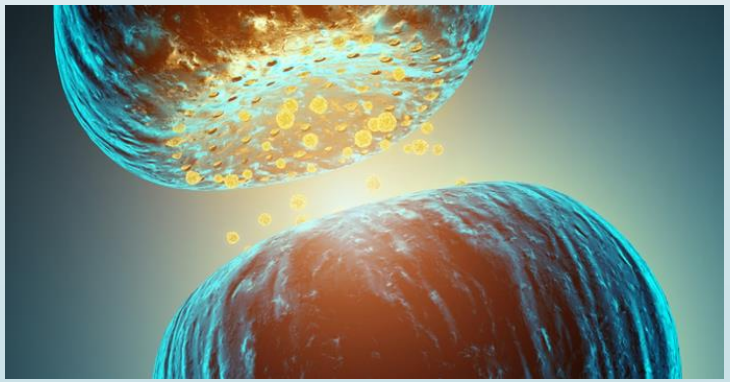
Matlab w służbie neuronauce

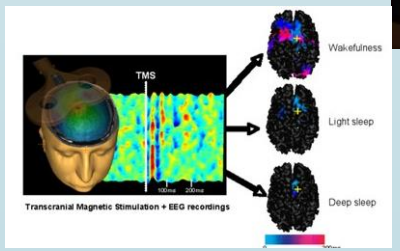
Czyli od podstaw Matlaba do własnoręcznego tworzenia ERPów

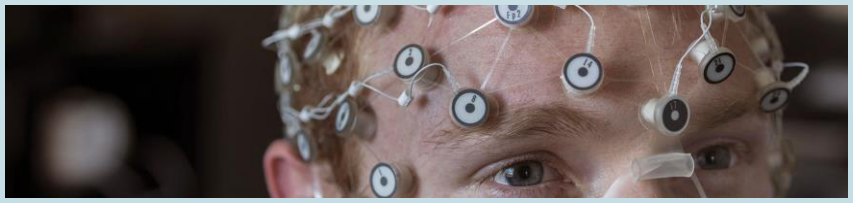
Dzisiaj nauczymy się podstaw Matlaba

- Cel jaki nam przyświeca – umieć własnoręcznie (bez interfejsu EEGlaba) stworzyć i zaprezentować na ekranie ERPa
- Wykorzystamy też do tego funkcje EEGlaba
- Umiejętność własnoręcznego (tzn. z poziomu command window bądź skryptu) poddawania danych prostym przetworzeniom jest konieczna by przetrwać w neuronaukowej dżungli









Podstawowe operacje

- Dodajemy, odejmujemy itp.:

```
>> 12 + 5
ans =
    17

>> 2 * 11.5
ans =
    23
```

Podstawowe operacje

- Możemy dodawać nawiasy aby przekazać jaka ma być kolejność operacji:

```
>> 11 + 2 / 3
>> (11 + 2) / 3
```

- Podnosimy do kwadratu za pomocą „czapeczki”:

```
>> 5.25 ^ 2
>> ( (17 + 4) / 8) ^ 2
```

- Aby wyciągnąć pierwiastek używamy funkcji `sqrt()` (więcej o funkcjach w drugiej części zajęć):

```
>> sqrt(123)
```

Zmienne

- Aby przechowywać wyniki przeróżnych operacji używamy **zmiennych**:

```
>> MojaPierwszaZmienna = 4 + 3
>> zmnn = 12 - 7
```

- Używamy średnika (;) aby stłumić gadatliwość Matlaba:

```
>> moja_fajna_zmienna = 1 + 8 + 5 + 9;
```

- Nazwy zmiennych muszą zaczynać się od litery oraz nie mogą zawierać innych znaków niż:
  [a-z] [0-9] _
  (klamry i myślniki grupują tu tylko znaki, nazwy zmiennych nie mogą ich zawierać)

Operacje na zmiennych

- Na zmiennych można przeprowadzać rozmaite operacje:

```
>> Wombaty = 523;
>> km = 11;
>> Wombat_na_km = Wombaty / km;
```

- Zmienne nie muszą przyjmować wartości liczbowej, mogą być też tekstowe:

```
>> nazwa_pliku = 'wulgarny wombat.set';
```

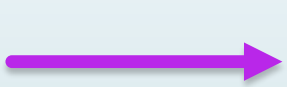
- Mogą też być np. wektorami (uporządkowanymi listami liczb):

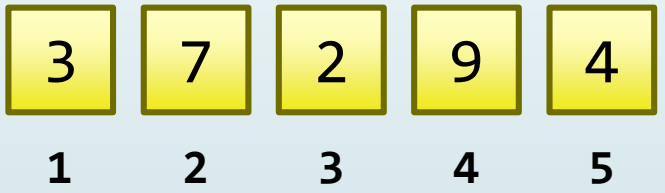
```
>> najlepszy_wektor = [3, 7, 2, 9, 4];
```

Wektory i adresowanie [1/8]

- Wektory, jedne z najczęściej używanych formatów danych w Matlabie, to po prostu sekwencje liczb:

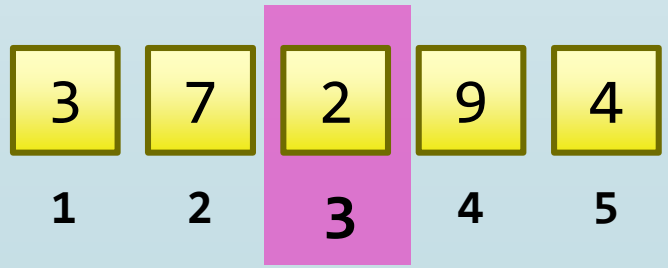
```
>> najlepszy_wektor = [3, 7, 2, 9, 4];
```

najlepszy_wektor 



- Możemy więc prosić o podanie wartości konkretnego elementu w sekwencji:

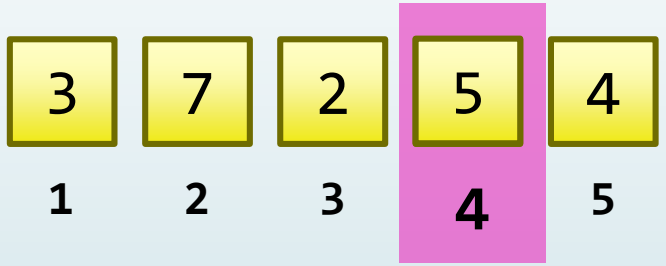
```
>> najlepszy_wektor(3)
```



Wektory i adresowanie [2/8]

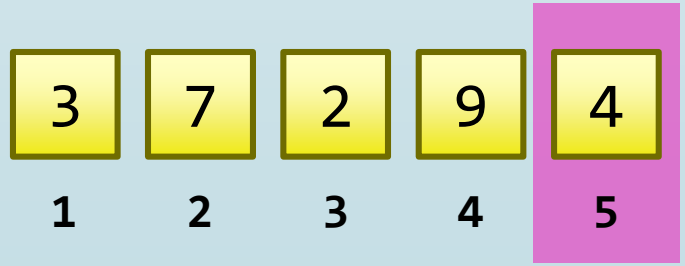
- Możemy też w ten sposób zmieniać wartości dla konkretnego elementu wektora:

```
>> najlepszy_wektor(4) = 5;
```



- Możemy też adresować od końca:

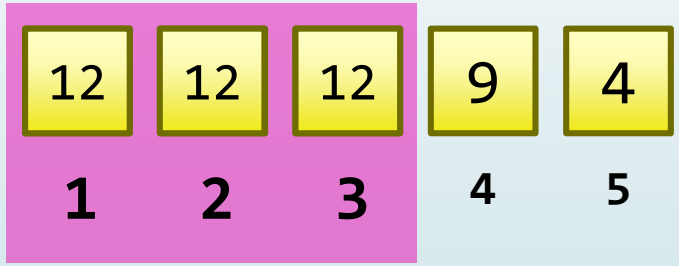
```
>> najlepszy_wektor(end - 3)
```



Wektory i adresowanie [3/8]

- Jeżeli chcemy dostać się do wielu elementów na raz możemy zaadresować wektor wektorem:

```
>> najlepszy_wektor([1, 2, 3]) = 12;
```



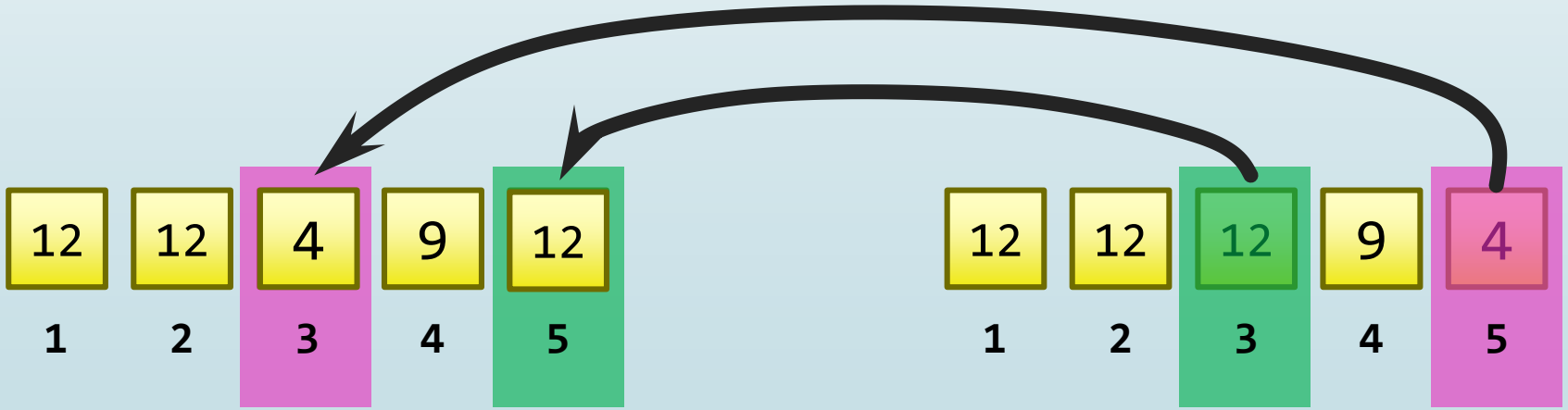
- Jeżeli chcemy w ten sposób przyporządkować różnym elementom wektora różne wartości *długość wektora adresującego oraz długość wektora nowych wartości muszą się zgadzać*:

```
>> najlepszy_wektor([2, 4, 5]) = [3, 8, 6];
```

Wektory i adresowanie [4/8]

- W ten sam sposób możemy np. zamienić elementy wektora miejscami:

```
>> najlepszy_wektor([3, 5]) = ...
       najlepszy_wektor([5, 3]);
```



Wektory i adresowanie [5/8]

- Zadanie dla Was, Młodzi Neurokognitywiści:

Stwórz wektor o nazwie neuron składający się sekwencji liczb: 6, 9, 3, 4, 1, 2, 3, 3, 7

Zmień trzy ostatnie elementy na 23 aby sekwencja wyglądała tak: 6, 9, 3, 4, 1, 2, 23, 23, 23

Spróbuj podmienić co drugi element wektora neuron na sekwencję liczb: 4, 3, 0, 1
tak aby neuron miał teraz postać:
6, 4, 3, 3, 1, 0, 23, 1, 23