# Wstęp do Programowania
# potok funkcyjny

Marcin Kubica
kubica@mimuw.edu.pl

2018/2019

# Outline

# Czym jest magia?

Czym jest programowanie?
Zobaczymy, że jest ono dziedziną **magii**!
Ale czym jest magia?

**Definition (Wikipedia en/pl)**

Magia — wykorzystanie rytuałów, symboli, [. . . ] języka i zaklęć, w celu wykorzystania nadnaturalnych sił do kształtowania rzeczywistości.

# Czym jest magia?

Czym jest programowanie?
Zobaczymy, że jest ono dziedziną **magii**!
Ale czym jest magia?

**Definition (Wikipedia en/pl)**

Magia — wykorzystanie rytuałów, symboli, [. . . ] języka i zaklęć, w celu wykorzystania nadnaturalnych sił do kształtowania rzeczywistości.

# Czym jest magia? c.d.

## Definition (Wikipedia en/pl)

Typowe cechy praktyk magicznych:

- Rytuały (np. logowanie, oddawanie programów zaliczeniowych),
  wykonywane z użyciem specjalnych obiektów (np. indeks),
  w określonym miejscu i czasie (np. godziny otwarcia dziekanatu).
- Osoba, przedmioty i miejsce wykonywania rytuału mogą wymagać zachowania czystości (np. zakaz jedzenia w laboratorium).
- Magiczne symbole (np. $T(n) = \Theta(n \cdot \log n)$).
- Język magiczny, zwykle archaiczny i cokolwiek dziwaczny. Ocaml!

# Czym jest magia? c.d.

## Definition (Wikipedia en/pl)

Typowe cechy praktyk magicznych:

- Rytuały (np. logowanie, oddawanie programów zaliczeniowych), wykonywane z użyciem specjalnych obiektów (np. indeks), w określonym miejscu i czasie (np. godziny otwarcia dziekanatu).
- Osoba, przedmioty i miejsce wykonywania rytuału mogą wymagać zachowania czystości (np. zakaz jedzenia w laboratorium).
- Magiczne symbole (np. $T(n) = \Theta(n \cdot \log n)$).
- Język magiczny, zwykle archaiczny i cokolwiek dziwaczny. Ocaml!

# Czym jest magia? c.d.

## Definition (Wikipedia en/pl)

Typowe cechy praktyk magicznych:

- Rytuały (np. logowanie, oddawanie programów zaliczeniowych),
  wykonywane z użyciem specjalnych obiektów (np. indeks),
  w określonym miejscu i czasie (np. godziny otwarcia dziekanatu).
- Osoba, przedmioty i miejsce wykonywania rytuału mogą wymagać zachowania czystości (np. zakaz jedzenia w laboratorium).
- Magiczne symbole (np. $T(n) = \Theta(n \cdot \log n)$).
- Język magiczny, zwykle archaiczny i cokolwiek dziwaczny. Ocaml!

# Czym jest magia? c.d.

## Definition (Wikipedia en/pl)

Typowe cechy praktyk magicznych:

- Rytuały (np. logowanie, oddawanie programów zaliczeniowych),
  wykonywane z użyciem specjalnych obiektów (np. indeks),
  w określonym miejscu i czasie (np. godziny otwarcia dziekanatu).
- Osoba, przedmioty i miejsce wykonywania rytuału mogą wymagać zachowania czystości (np. zakaz jedzenia w laboratorium).
- Magiczne symbole (np. $T(n) = \Theta(n \cdot \log n)$).
- Język magiczny, zwykle archaiczny i cokolwiek dziwaczny. Ocaml!

# Programy jako zaklęcia magiczne

- Programy to zaklęcia zapisane w specjalnym języku (tzw. języku programowania)
- Programy działają na **procesy obliczeniowe**.
- Procesy obliczeniowe to rodzaj **duchów** mieszkających w komputerach.
- Czy na pewno? A czym są duchy?

# Procesy obliczeniowe jako duchy

## Definition (Duchy)

Charakterystyka duchów:

- Niematerialne.
- Potencjał intelektualny.
- Zdolność do komunikowania się ze światem realnym.
- Moc sprawcza.

Procesy obliczeniowe posiadają wszystkie te cechy!

# Kilka podstawowych pojęć

| | |
|---|---|
| Program | zaklęcie zapisane w języku programowania. |
| Programowanie | formułowanie programu mającego osiągnąć zamierzony cel. |
| Weryfikacja | sprawdzenie, że program osiąga zamierzony cel. |
| Testowanie | szukanie kontr-przykładu, że program nie osiąga zamierzonego celu. |
| Algorytm | sposób działania procesu obliczeniowego wyrażony przez program. |

$$program = algorytm + język\ programowania$$

# Muhammad Ibn Mūsā al-Khwārizmi (ok. 780–850 r. n.e.)

- Perski matematyk, geograf i astronom działający w *Domu Mądrości* za czasów kalifa Al-Ma'mūna.
- Przyczynił się do upowszechnienia zapisu pozycyjnego i cyfr „arabskich".
- Autor m.in. „Zasad redukcji i przenoszenia" (Kitāb al-jabr wa'l-muqābala), od której pochodzi słowo *algebra*.
- Od jego nazwiska pochodzi słowo **algorytm** (al-Khwārizmi, Al-Chuwarizmi, Alchwarizmi, Algorithmi, Algorithmus).

# Metody rozwiązywania równań kwadratowych

- Liczby utożsamiamy tu z długościami odcinków.
- Mnożenie odpowiada powierzchni prostokątów.
- W tych czasach pojęcie liczb ujemnych i zera nie było jeszcze znane.

| $a$ | $b$ | $c$ | równanie | przykład | metoda |
|---|---|---|---|---|---|
| $+$ | $+$ | $+$ | $ax^2 + bx + c = 0$ | — | — |
| $+$ | $+$ | 0 | $ax^2 + bx = 0$ | — | — |
| $+$ | $+$ | $-$ | $ax^2 + bx = c$ | $x^2 + 10x = 39$ | metoda 1 |
| $+$ | 0 | $+$ | $ax^2 + c = 0$ | — | — |
| $+$ | 0 | 0 | $ax^2 = 0$ | — | — |
| $+$ | 0 | $-$ | $ax^2 = c$ | $5x^2 = 80$ | metoda 2 |
| $+$ | $-$ | $+$ | $ax^2 + c = bx$ | $x^2 + 21 = 10x$ | metoda 3 |
| $+$ | $-$ | 0 | $ax^2 = bx$ | $x^2 = 5x$ | metoda 4 |
| $+$ | $-$ | $-$ | $ax^2 = bx + c$ | $x^2 = 3x + 4$ | metoda 5 |

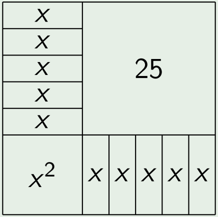
# Metody rozwiązywania równań kwadratowych

- Liczby utożsamiamy tu z długościami odcinków.
- Mnożenie odpowiada powierzchni prostokątów.
- W tych czasach pojęcie liczb ujemnych i zera nie było jeszcze znane.

| $a$ | $b$ | $c$ | równanie | przykład | metoda |
|---|---|---|---|---|---|
| + | + | + | $ax^2 + bx + c = 0$ | — | — |
| + | + | 0 | $ax^2 + bx = 0$ | — | — |
| + | + | − | $ax^2 + bx = c$ | $x^2 + 10x = 39$ | metoda 1 |
| + | 0 | + | $ax^2 + c = 0$ | — | — |
| + | 0 | 0 | $ax^2 = 0$ | — | — |
| + | 0 | − | $ax^2 = c$ | $5x^2 = 80$ | metoda 2 |
| + | − | + | $ax^2 + c = bx$ | $x^2 + 21 = 10x$ | metoda 3 |
| + | − | 0 | $ax^2 = bx$ | $x^2 = 5x$ | metoda 4 |
| + | − | − | $ax^2 = bx + c$ | $x^2 = 3x + 4$ | metoda 5 |

# Metoda 1

## Example ($x^2 + 10x = 39$)




Pole dużego kwadratu wynosi $39 + 25 = 64 = 8 \times 8$.
Stąd $x + 5 = 8$, czyli $x = 3$.

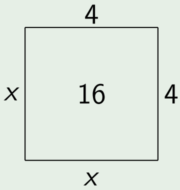
# Zdziwienie dnia

## Zdziwienie dnia

W metodzie rozwiązywania równań kwadratowych Al-Chwarizmiego wykorzystywane jest pierwiastkowanie.
Równocześnie liczby są utożsamiane z długościami odcinków lub powierzchniami figur.
Jak wyznaczyć pierwiastek z długości odcinka używając cyrkla, linijki i odcinka jednostkowego?

# Metoda 2

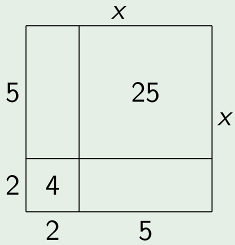
**Example ($5x^2 = 80$)**




$5x^2 = 80$, stąd $x^2 = \frac{80}{5} = 16$, czyli $x = 4$.

# Metoda 3

## Example ($x^2 + 21 = 10x$)




Z kwadratu $x \times x$ wycinamy dwa prostokąty $5 \times x$.
Pokrywają się one w kwadracie $5 \times 5$.
Mały kwadracik ma powierzchnię
$x^2 - 10x + 25 = x^2 - 10x + 21 + 4 = 4 = 2^2$.
$x = 5 + 2 = 7$.

# Metoda 4

**Example ($x^2 = 5x$)**

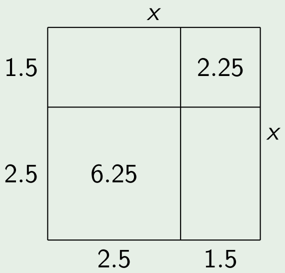
Równanie $ax^2 = bx$ ma dwa rozwiązania:

- $x = 0$,
- $x = \frac{b}{a}$

Dla $x^2 = 5x$ mamy $x = 0$ lub $x = 5$.

# Metoda 5

## Example ($x^2 = 3x + 4$)




Z kwadratu $x \times x$ wycinamy dwa prostokąty $1.5 \times x$
Pokrywają się one w kwadracie $1.5 \times 1.5$.
Mały kwadracik ma powierzchnię
$x^2 - 3x + 1.5 \cdot 1.5 = 4 + 2.25 = 6.25 = 2.5^2$.
Stąd $x = 1.5 + 2.5 = 4$.

# Euklides

- **Euklides** ($E\acute{\upsilon}\kappa\lambda\varepsilon\acute{\iota}\delta\eta\sigma$) — grecki filozof i matematyk, przełom IV i III w p.n.e., Aleksandria.
- *Elementy* — kompendium wiedzy matematycznej, geometria, ale nie tylko.
- Algorytm Euklidesa — najstarszy (chyba) znany algorytm, największy wspólny dzielnik dwóch liczb.

# Euklides

- **Euklides** ($E\acute{\upsilon}\kappa\lambda\varepsilon\acute{\iota}\delta\eta\sigma$) — grecki filozof i matematyk, przełom IV i III w p.n.e., Aleksandria.
- *Elementy* — kompendium wiedzy matematycznej, geometria, ale nie tylko.
- Algorytm Euklidesa — najstarszy (chyba) znany algorytm, największy wspólny dzielnik dwóch liczb.

# Euklides

- **Euklides** ($E\acute{\upsilon}\kappa\lambda\varepsilon\acute{\iota}\delta\eta\sigma$) — grecki filozof i matematyk, przełom IV i III w p.n.e., Aleksandria.
- *Elementy* — kompendium wiedzy matematycznej, geometria, ale nie tylko.
- Algorytm Euklidesa — najstarszy (chyba) znany algorytm, największy wspólny dzielnik dwóch liczb.

# Algorytm Euklidesa

Wersja z odejmowaniem, $NWD(x, y) = NWD(x - y, y)$.

**Algorytm**

1. Dopóki $x \neq y$ powtarzaj następujący krok:
2. Od większej z liczb $x$ i $y$ odejmij mniejszą.
3. Wynik to $x = y = NWD(x, y)$.

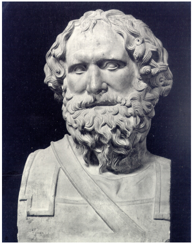
# Algorytm Euklidesa

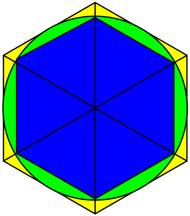
Wersja z odejmowaniem, $NWD(x, y) = NWD(x - y, y)$.

## Algorytm

1. Dopóki $x \neq y$ powtarzaj następujący krok:
2. Od większej z liczb $x$ i $y$ odejmij mniejszą.
3. Wynik to $x = y = NWD(x, y)$.

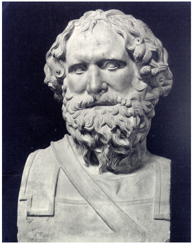
# Archimedes

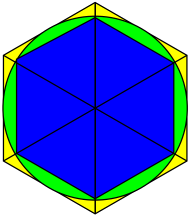




- **Archimedes** (*Άρχιμήδησ*, ok. 287–212 p.n.e.) wszechstronny grecki filozof, Syrakuzy.
- Opracował algorytm przybliżania liczby $\pi$.
- Okrąg jednostkowy z opisanym i wpisanym $n$-kątem foremnym.
- Proste oszacowanie dla sześciokąta: $3 < \pi < 2\sqrt{3}$

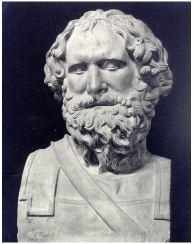
# Archimedes

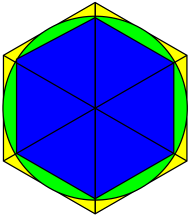




- **Archimedes** ($\acute{A}\rho\chi\iota\mu\acute{\eta}\delta\eta\sigma$, ok. 287–212 p.n.e.) wszechstronny grecki filozof, Syrakuzy.
- Opracował algorytm przybliżania liczby $\pi$.
- Okrąg jednostkowy z opisanym i wpisanym $n$-kątem foremnym.
- Proste oszacowanie dla sześciokąta: $3 < \pi < 2\sqrt{3}$

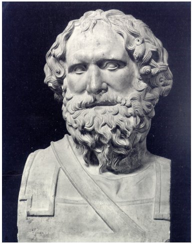
# Archimedes

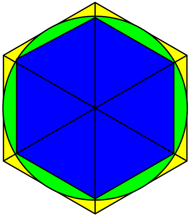




- **Archimedes** ($\acute{A}\rho\chi\iota\mu\acute{\eta}\delta\eta\sigma$, ok. 287–212 p.n.e.) wszechstronny grecki filozof, Syrakuzy.
- Opracował algorytm przybliżania liczby $\pi$.
- Okrąg jednostkowy z opisanym i wpisanym *n*-kątem foremnym.
- Proste oszacowanie dla sześciokąta: $3 < \pi < 2\sqrt{3}$

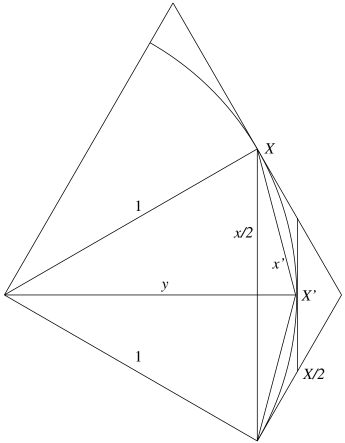
# Archimedes





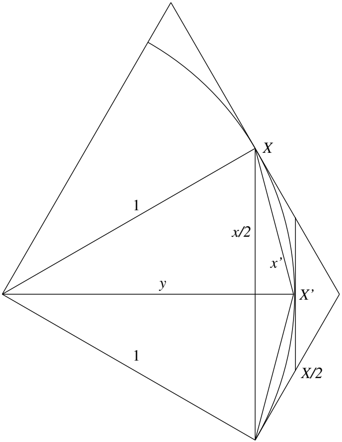
- **Archimedes** ($\acute{A}\rho\chi\iota\mu\acute{\eta}\delta\eta\sigma$, ok. 287–212 p.n.e.) wszechstronny grecki filozof, Syrakuzy.
- Opracował algorytm przybliżania liczby $\pi$.
- Okrąg jednostkowy z opisanym i wpisanym *n*-kątem foremnym.
- Proste oszacowanie dla sześciokąta: $3 < \pi < 2\sqrt{3}$

# Algorytm Archimedesa




- $X$ i $x$ — boki $n$-kątów foremnych
  $X'$ i $x'$ — boki $2n$-kątów foremnych
- Oznaczmy: $y = \sqrt{1 - \frac{x^2}{4}}$
- $X = \frac{x}{y} = \frac{2x}{\sqrt{4-x^2}}$

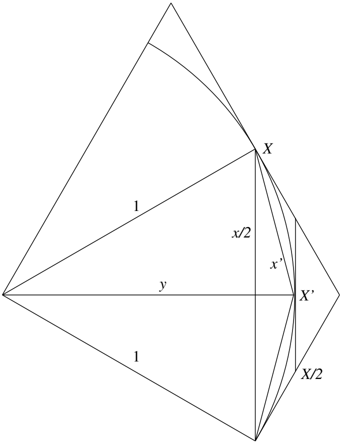
# Algorytm Archimedesa, c.d.




$$y = \sqrt{1 - \frac{x^2}{4}} \qquad X = \frac{2x}{\sqrt{4 - x^2}}$$

$$\begin{aligned}
x' &= \sqrt{\frac{x^2}{4} + (1-y)^2} = \\
&= \sqrt{\frac{x^2}{4} + 1 + y^2 - 2y} = \\
&= \sqrt{\frac{x^2}{4} + 1 + 1 - \frac{x^2}{4} - 2\sqrt{1 - \frac{x^2}{4}}} = \\
&= \sqrt{2 - \sqrt{4 - x^2}} = \sqrt{2 - \frac{2x}{X}}
\end{aligned}$$

# Algorytm Archimedesa, c.d.




$$
\begin{aligned}
x' &= \sqrt{2 - \frac{2x}{X}} \\
X' &= \frac{2x'}{\sqrt{4 - x'^2}} = \\
&= \frac{2\sqrt{2 - \frac{2x}{X}}}{\sqrt{4 - 2 + \frac{2x}{X}}} = \\
&= 2\sqrt{\frac{2 - \frac{2x}{X}}{2 + \frac{2x}{X}}} = 2\sqrt{\frac{X - x}{X + x}}
\end{aligned}
$$

# Algorytm Archimedesa, c.d.

- Otrzymaliśmy następujące zależności:
  $x' = \sqrt{2 - \frac{2x}{X}} \qquad X' = 2\sqrt{\frac{X-x}{X+x}}$
- Kolejne kroki algorytmu Archimedesa:

| $n$ | $x$ | $X$ | $\pi >$ | $\pi <$ |
|---|---|---|---|---|
| 6 | 1.0000000 | 1.1547005 | 3.0000000 | 3.4641016 |
| 12 | 0.5176380 | 0.5358983 | 3.1058285 | 3.2153903 |
| 24 | 0.2610523 | 0.2633049 | 3.1326286 | 3.1596599 |
| 48 | 0.1308062 | 0.1310869 | 3.1393502 | 3.1460862 |
| 96 | 0.0654381 | 0.0654732 | 3.1410319 | 3.1427145 |

# Algorytm Archimedesa, c.d.

- Otrzymaliśmy następujące zależności:
  $x' = \sqrt{2 - \frac{2x}{X}} \qquad X' = 2\sqrt{\frac{X-x}{X+x}}$
- Kolejne kroki algorytmu Archimedesa:

| $n$ | $x$ | $X$ | $\pi >$ | $\pi <$ |
|---|---|---|---|---|
| 6 | 1.0000000 | 1.1547005 | 3.0000000 | 3.4641016 |
| 12 | 0.5176380 | 0.5358983 | 3.1058285 | 3.2153903 |
| 24 | 0.2610523 | 0.2633049 | 3.1326286 | 3.1596599 |
| 48 | 0.1308062 | 0.1310869 | 3.1393502 | 3.1460862 |
| 96 | 0.0654381 | 0.0654732 | 3.1410319 | 3.1427145 |

# Języki programowania są wszędzie

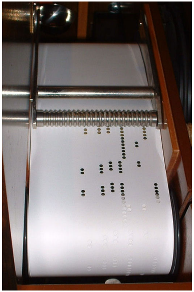
Programowanie było znane od dawna, choć nie wiedziano jeszcze, że jest to programowanie.

## Example

Przepisy kulinarne

- zamówienie — specyfikacja,
- przepis — program,
- gotowanie — wykonanie,
- potrawa — efekt,
- smakuje? — testowanie,
- weryfikacja — ?

# Języki programowania są wszędzie — muzyka



## Example

Zapis nutowy:

- nuty — program,
- katarynka, pianola — komputer przenośny, serwer,
- taśma dziurkowana — skompilowany program,
- muzyka — efekt działania katarynki, pianoli, czy orkiestry,

# Języki programowania są wszędzie — MetaFont

$\mathcal{W}$
$\mathcal{P}$
$\mathcal{F}$

**Example**

Opisy skalowalnych czcionek (np. MetaFont):

- opis czcionek — program,
- wygląd czcionek przy zadanej rozdzielczości, parametrach i dla zadanego urządzenia — wykonanie,
- wygląd czcionek — efekt.

# Języki programowania są wszędzie — systemy składu tekstu

## Example

Systemy składu tekstu (np. LaTeX, MS Word, Open Office) zawierają mechanizm **makrodefinicji**:

- tekst + makrodefinicje — program,
- skład tekstu — wykonanie,
- wydruk — efekt działania.

# Języki programowania są wszędzie — $\LaTeX$

## Example

Definicja tabelki, która sama się wypełnia liczbami Fibonacciego.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| $Fib_i$ | 1 | 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 | 89 | 144 |

```
\newcounter{i}
\newcounter{a}
\newcounter{b}
\newcounter{c}
\newcommand{\fibtab}[1]{
  \def\heads{|l|}
  \def\inds{$i$}
  \def\wyniki{$Fib_i$}
  \setcounter{i}{0}
  \setcounter{a}{0}
  \setcounter{b}{1}
  \@whilenum\value{i}<#1\do {
    \addtocounter{i}{1}
    \edef\heads{\heads l|}
    \edef\inds{\inds & \thei}
    \edef\wyniki{\wyniki & \theb}
    \setcounter{c}{\value{a}}
    \addtocounter{c}{\value{b}}
    \setcounter{a}{\value{b}}
    \setcounter{b}{\value{c}}
  }
  \begin{tabular}{\heads}
  \hline \inds \\
  \hline \wyniki \\
  \hline
  \end{tabular}
}
\fibtab{12}
```

# Dziedzina algorytmiczna

## Definition

*Dziedzina algorytmiczna*, to zestaw elementarnych pojęć, których możemy używać opisując algorytmy:

- zbiory wartości,
- funkcje i stałe,
- relacje.

Dziedzina algorytmiczna jest „wbudowana” w język programowania. Możemy jednak mówić o niej niezależnie od języka programowania.

# Dziedzina algorytmiczna, c.d.

## Example

Dziedzina algorytmiczna al-Khwārizmiego:

- $\mathbb{R}_+$,
- $1$, $+$, $-$, $\times$, $/$, $\sqrt{\ }$,
- $<$, $=$.

Jeśli mamy zbiór wartości logicznych { prawda, fałsz }, to relacje możemy utożsamić z funkcjami w zbiór wartości logicznych.

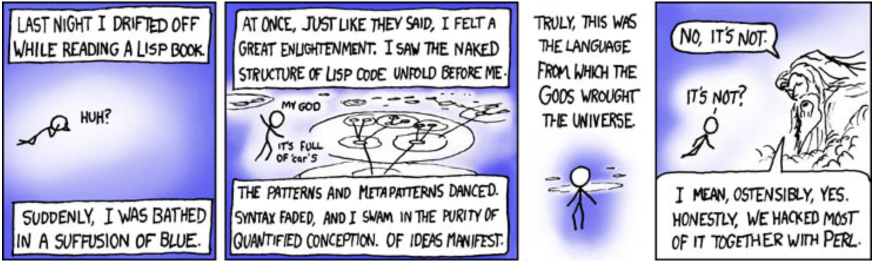
# Zasady zaliczenia

- Kolokwia
- Programy zaliczeniowe
- Zwolnienia z egzaminu

# Materiały

- Skrypt
  `http://www.mimuw.edu.pl/~kubica/wpf/`
- Moodle
  `https://moodle.mimuw.edu.pl/`
- Slajdy
  `http://smurf.mimuw.edu.pl/drupal6/?q=wstep_do_programowania_funkcyjny`
- Literatura:
  - Y. Minsky, A. Madhavapeddy, J. Hickey, *Real World OCaml*, O'Reilly 2014.
  - J. Tomasiewicz, *Zaprzyjaźnij się z algorytmami*, PWN 2017.
  - A. Hunt, D. Thomas, *Pragmatyczny programista*, WNT 2009.
  - H. Abelson, G. J. Sussman, *Struktura i interpretacja programów komputerowych*, WNT 2002.

# Deser




http://xkcd.com/224/