



[dimon.work/kurs.html](http://dimon.work/kurs.html)

# 1. Oprogramowanie interaktywne

# XEROX 6085 Workstation

## User-Interface Design

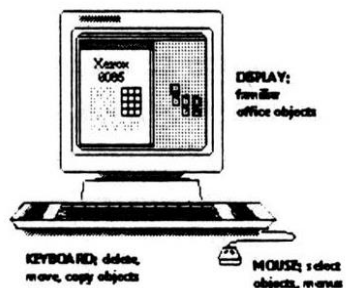
To make it easy to compose text and graphics, to do electronic filing, printing, and mailing all at the same workstation, requires a revolutionary user interface design.

**Bit-map display** - Each of the pixels on the 19" screen is mapped to a bit in memory; thus, arbitrarily complex images can be displayed. The 6085 displays all fonts and graphics as they will be printed. In addition, familiar office objects such as documents, folders, file drawers and in-baskets are portrayed as recognizable images.

**The mouse** - A unique pointing device that allows the user to quickly select any text, graphic or office object on the display.

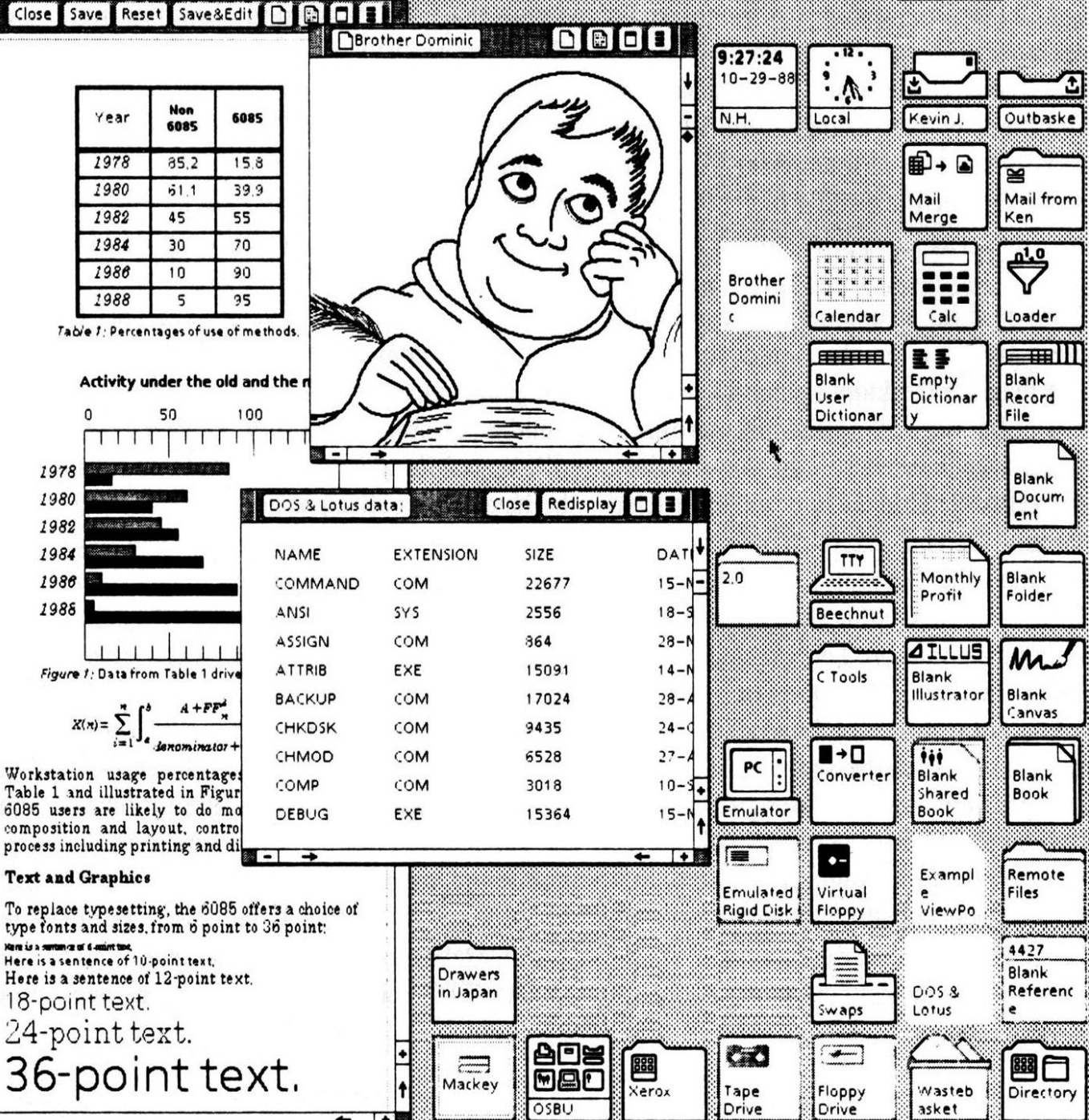
## See and Point

All functions are visible to the user on the keyboard or on the screen. The user does filing and retrieval by selecting them with the mouse and touching the MOVE, COPY, DELETE or PROPERTIES command keys. Text and graphics are edited with the same keys.



## Shorter Production Times

Experience at Xerox with prototype work stations has shown shorter production times and thus lower costs, as a function of the percentage of use of the workstations. The following equation can be used to express this:



**Skeumorfizm** - ornament lub element designu, który jest kopiowany z formy innego obiektu, ale wykonany z innych materiałów lub inną metodą. /Википедия.



Elementy interfejsu pierwszych wersji iOS.

**Skeumorfizm** - ornament lub element designu, który jest kopiowany z formy innego obiektu, ale wykonany z innych materiałów lub inną metodą [/Википедия](#).



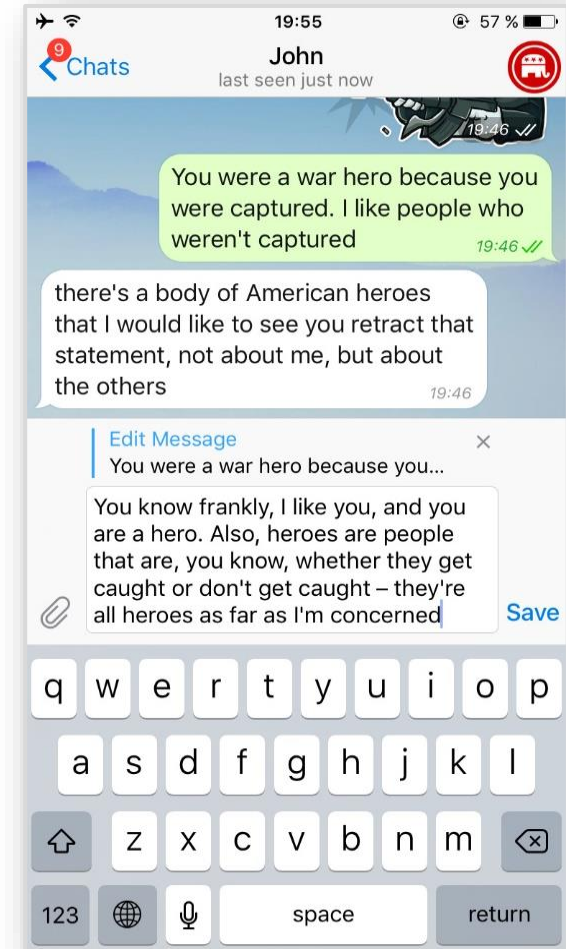
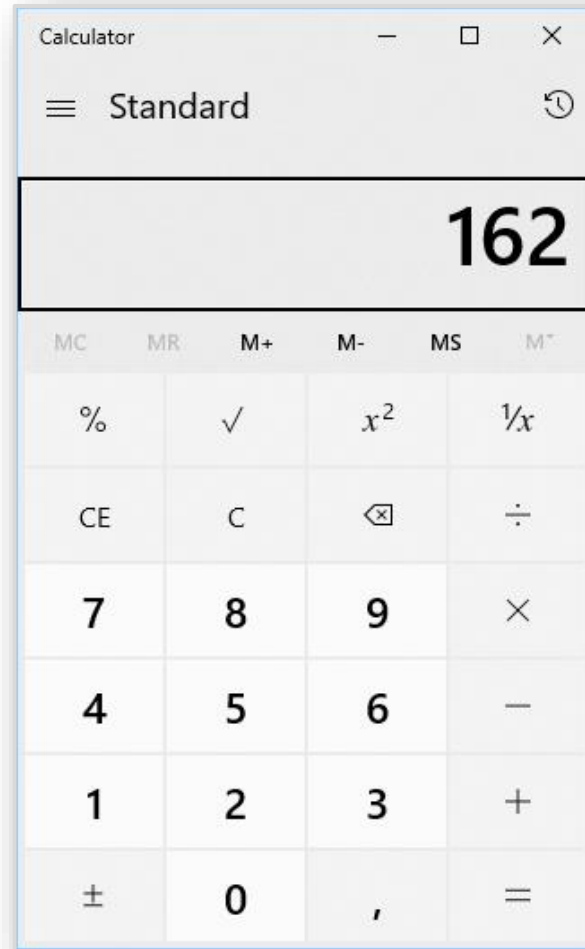
# Opragromawanie interaktywne



*Zdarzeniowy model sterowania*

[https://en.wikipedia.org/wiki/Event-driven\\_programming](https://en.wikipedia.org/wiki/Event-driven_programming)

# Oprogramowanie interaktywne

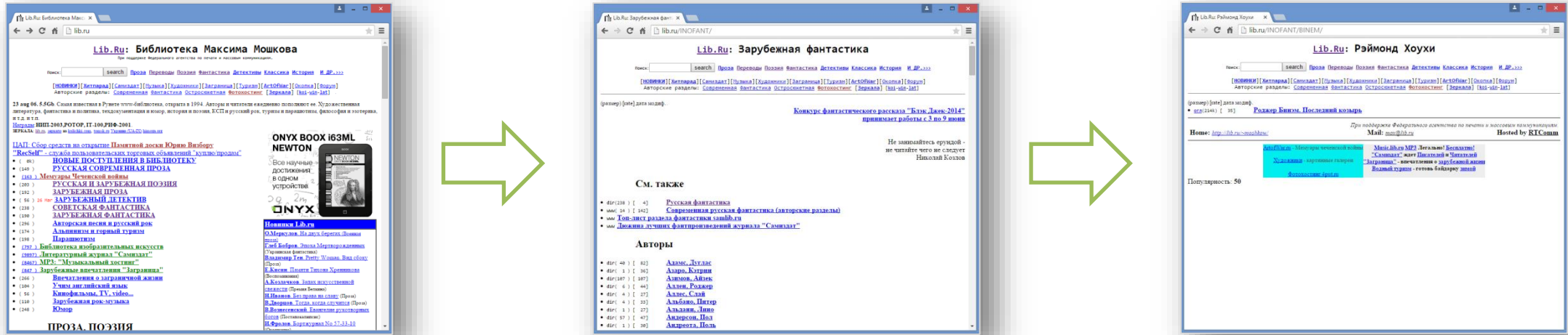


## 2. Oprogramowanie interaktywne

### ...w przedtądarce...

# HTML jest statyczny.

Nie jest to zaskakujące, ponieważ **HTML** (i **CSS**) nie jest językiem programowania.



Po załadowaniu strony do przeglądarki pozostaje ona niezmienną. Aby uzyskać inną informację, należy załadować nową stronę. Jednak użytkownicy (przyzwyczajeni do pracy z oprogramowaniem desktopowym) oczekują pewnej **interaktywności**.



# HTML/CSS – języki deklaratywne

**Deklaratywne** języki pozwalają za pomocą dyrektyw wskazać, jaki efekt chcemy osiągnąć, ale nie sposób, w jaki do tego dojść (komputer ustala sposób realizacji).  
Efekt uboczny: to, czego nie przewidują dostępne dyrektywy, nie może zostać zrealizowane.

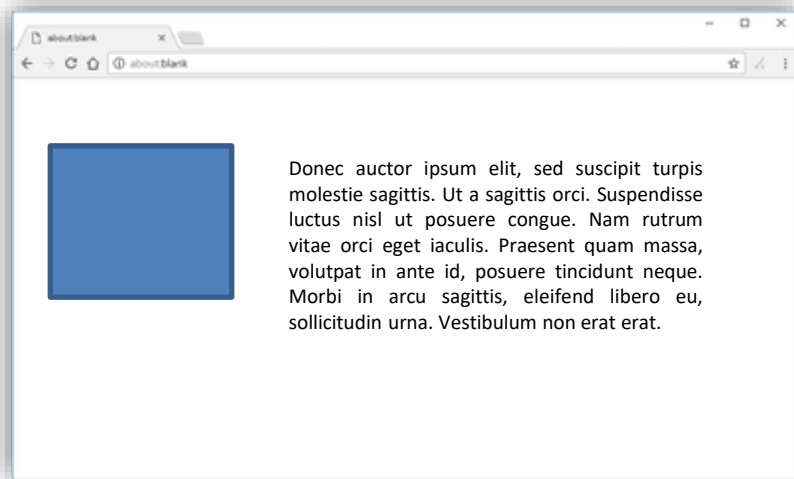
# JavaScript – język imperatywny

**Imperatywne** języki (zazwyczaj należą do nich języki programowania) składają się z instrukcji (małych działań, „kroków”) wykonywanych w określonej kolejności.

Kod w języku imperatywnym opisuje sposób osiągnięcia oczekiwanego wyniku.

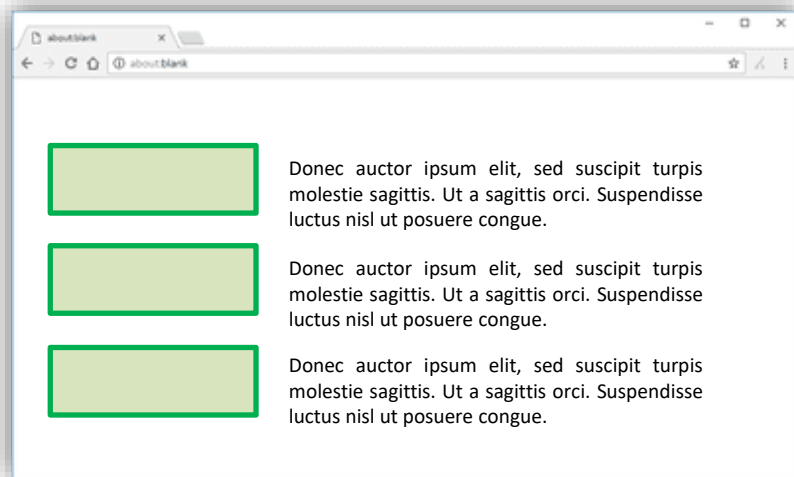
**Charakterystyczna cecha języków imperatywnych:** praca z danymi, opis logiki działania programu.

# Zadania JavaScript?



Zmiana zawartości  
strony (głównie  
danych).

A znaczniki są  
odpowiedzialne za  
zawartość strony...



# Zadania JavaScript

1. Manipulowanie elementami strony HTML (znacznikami)  
*(gdy strona znajduje się już w przeglądarce użytkownika);*

*A dokładniej: modyfikować znaczniki dokumentu. Ponieważ przeglądarka „narysuje” tylko to, co jest opisane w znacznikach. .*

2. Udzielać odpowiedzi na działania użytkownika  
*(reagować na działania użytkownika).*

# 3. W praktyce



Hello, JS!



Click on the photo to display the next one

Photo #1

## zdjęćGaleria

Zadanie **JavaScript** – zmieniać strukturę stronę, link do obrazu w znaczniku

`<img src='...'>` również należy do struktury strony ...

Skorzystaj z szablonu w repozytorium lekcji  
[dimon.work/kurs/javascript-demo-example/](https://github.com/dimon.work/kurs-javascript-demo-example/)

## 4. JavaScript – język programowania

# HTML



# CSS



# JS



**Trzy podstawowe technologie Front-Endu:**

# JavaScript – język programowania

*Jego «elementy składowe»*

**Zmienne / Typy / Operacje**

**Wstawienie (Instrukcje warunkowe)**

**Pętle / Tablice (struktury danych)**

**Funkcje**

**Obiekty**

# 5. Algorytm



# Algorytm

**Zadanie :** Skrypt powinien obliczać, ile PLN dziennie przynosi użytkownikowi depozyt ulokowany na półtora roku przy rocznej stopie procentowej 4.5%.

## Problemy :

- ✓ *Dostępne dane są niepełne lub część danych jest ukryta, trzeba je doprecyzować.*
- ✓ *Część danych jest nadmiarowa.*
- ✓ *Istnieją czynniki zewnętrzne, które mogą wpłynąć na wynik, ale nie są znane z góry.*

# Algorytmy

**Zadanie :** Skrypt powinien obliczać, ile PLN **dziennie** przynosi użytkownikowi depozyt ulokowany na **półtora roku** przy rocznej stopie procentowej **4.5%**.

## Algorytm:

1. Doprecyzowanie kwoty depozytu.

2. Obliczenie rocznego dochodu:

$$\text{Dochód} = \text{Kwota} * (4.5\% / 100);$$

3. Obliczenie dziennego dochodu:

$$\text{Dochód\_dzień} = \text{Dochód} / \text{Liczba\_dni\_w\_roku};$$

4. Obliczenie podatku:

$$\text{Kwota\_podatku} = \text{Dochód\_dzień} * ((19\% + 1,5\%) / 100);$$

5. Uwzględnienie podatku:

$$\text{Dochód\_dzień\_po\_podatku} = \text{Dochód\_dzień} - \text{Kwota\_podatku};$$

6. Wyświetlenie wyników.