Augmenting the human

Welcome!

This class is going to take a look at some of the original visions of the people who worked on some of the first digital technologies. Starting from there, we will trace the history of information technologies until today.

What did they envision as they were just scratching the surface of what computers could do? How does this vision carry today? Can we, and should we try to change the direction in which these technologies are going?

The second part of the class will be a review of HTML and introducing additional CSS concepts.

—

Housekeeping

Readings presentations:

- Session 3 - Maya
- Session 4 - Lucy
- Session 6 - Hong
- Session 10 - Reade

Transmediale / CTM / Berlinale

—

Plan for the day

- dreams of computers
- dreams of networks
- html review
- css styles

—

Dreams of computers

—

how would you qualify (y)our relationship with computers today?

—

when was the last time you used a computer to think deeply about a problem?

what are some future developments you're excited about? what about developments you're suspicious of?

—

J.C.R Licklider, *Man-Computer Symbiosis*, 1960.

J.C.R. Licklider was the director of DARPA, the research and development branch of the U.S. military. At the time, he was overseeing the ARPANet, which would eventually become the Internet (The Computer as a Communication Device). Licklider selected and directed most of the people who worked on the ARPAnet, the precursor to the internet. Working on psychoacoustics, he realized that most of his time was taken up by monotonous, repetitive tasks, which could be automated by a computer. In this piece of writing, he anticipates that the computer will be of great use to humans, by getting rid of the annoying and superfluous tasks.

What is an annoying and superfluous task remains debated, though. Talking about augmenting human capabilities, it's important to remember that computers can be—as Licklider wrote— a means to save us time from the mechanical tasks that take up most of our attention. The following question, then, is ***what do we do with the time we've gained?****.*

For example, searching through datasets in a library can result in more time dedicated to actual thinking about how those resources relate to us. Automatic formatting on a word processor allows us to take more time to think about the content.

In the end, taking care of formal details actually helps us narrow down what it is that we're doing, and even asking why we're going it.

To do that, he says computers would need to have better speed, better storage, better input, and better language.

Due to Moore's law, which observes that computers double in speed and storage every two years (see the graph here), storage and speed are no longer a problem.

With our contemporary screens, keyboards, mouses and interfaces, input systems are no longer a problem.

Still, remains the problem of the language, of expressing what we want the computer to do.

This particular reading is interesting in how he lists out the practical obstacles of reaching such a collaboration (speed, language, and Input/Output). Out of those three obstacles, two of them have been overcome a long time ago. It's only the language part that is the problem. So now that we are close to having gotten rid of big technical hurdles, what are the non-technical hurdles left?

—

David Rokeby, *The Computer as a prosthetic organ for philosophy*, 2003.

David Rokeby is a digital artist, known for very nervous system , one of the first interactive computer installations.

His argument is that using a computer to do non-computer things (sensing a human dancing, for instance) forces us to really think about what is this thing that we are trying to represent.

—

how do you get a computer to give you the meaningful images of your life?

Turns out, it might seem a bit tricky at first, but one might be able to get close!

—

Epistemic actions

—

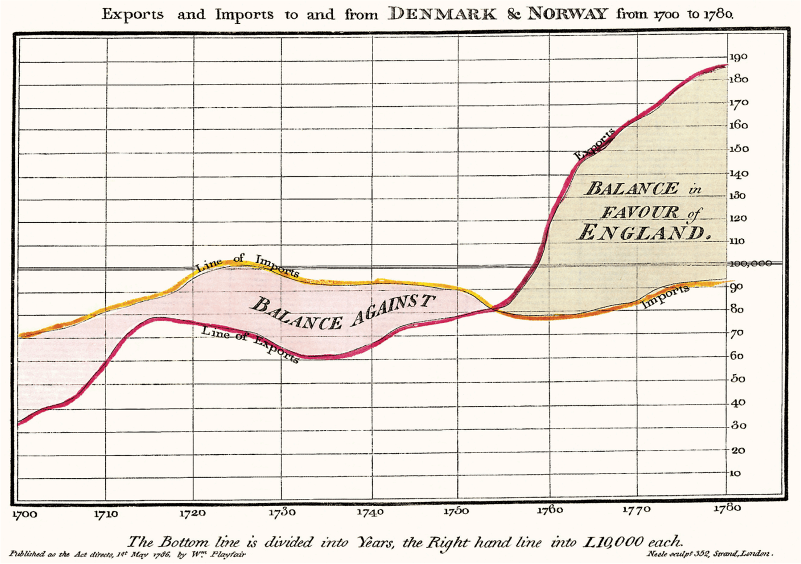
they can help us access answers to known problems, by doing **mechanical** tasks (finding, filtering, sorting, indexing).

*Licklider approaches the problem in terms of **cooperation**: what is it that we want to do? Do we want to solve math formulas, to pick a movie to watch, to invest money in different funds or to learn about a particular topic?*

In order to answer these questions, he compares the strengths of humans and the strengths of machines and extrapolates on how they could work together. This working together, this collaboration, implies that the answer to the problem is not yet known. There is no clear information that needs to be delivered, but rather information that needs to be pieced together in order for the investigator (i.e. the learner) to *create their own knowledge. Such an approach is in line with a pedagogical theory called **constructivism**, which we will talk more about in the week about and education.*

—

computers can help us formulate problems and *construct* answers by organizing and presenting information.



Data visualization reveals some truths

—

they did not answer the questions of **intent** (what do I want to do) and **purpose** (how do I want to do it).

—

to access the capabilities of the computer, we first need to **translate** the world into computer terms!

this involves **standardization** and **disambiguation**.

Standardization is a form of abstraction: you only keep the parts of something that fits the standard. The rest is discarded.

So in order to make the computer work with something that matters to us (generally, things we want to remember), then we first need to remove some parts that might be meaningful, and then we need to represent this data such that sense can be made from it when we (re-)access it via the computer.

In the beginning of computing, it was pretty easy: mathematics mean what they are. We're less sure about the creation of meaning than about the way differential equations work, so in a lot of computer-aided applications, the problem is still to formulate exactly what the problem is. But then problems got a different kind of complex. "I would like to see all that is relevant to the current task at hand" is too vague of a query for a computer to fulfill, and is a field that is still currently being worked on, particularly in the realm of personal assistants (e.g. Siri, Alexa, etc.). It's therefore still the task of the designer and developer to try to anticipate, accomodate, and encourage anything that their user might think about or attempt to do.

—

data > information > meaning

the **presentation** of information contributes to meaning

—

Ivan Sutherland, the next director of DARPA was thinking about *The Ultimate Display*

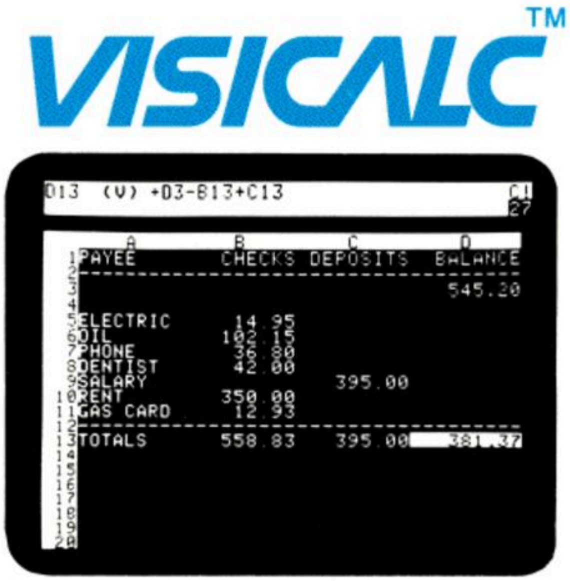
- the first touchscreen, SKETCHPAD
- the first virtual reality headset, SWORD OF DAMOCLES

One of the people Licklider influenced was on Doug Engelbart, a Stanford researcher who founded the Augmented Research Center, particularly well known for presenting the mother of all demos (video). Engelbart was also influenced by another thinker, Vanevar Bush, who wrote As We May Think .

—

displays adds spatiality as a way to **reveal** information (making visible an object), and to **lay it out** (making visible the relationship of objects) and to **manipulate it** (changing it in real-time to observe its behaviour).

—



```
D13  (V) +D3-B13+C13                              C!
                                                  27
          A            B         C          D
 1PAYEE            CHECKS  DEPOSITS   BALANCE
 2-----------------------------------------------
 3                                     545.20
 4
 5ELECTRIC          14.95
 6OIL              102.15
 7PHONE             36.80
 8DENTIST           42.00
 9SALARY                     395.00
10RENT             350.00
11GAS CARD          12.93
12-----------------------------------------------
13TOTALS           558.83    395.00    381.37
14
15
16
17
18
19
20
```

VisiCalc, the first spreadsheet software, completely changed how we see data

While Sutherland is focused on the wonderland of mathematics, it is interesting to see how he considers the display: not just as a means to show information, but also as a means to ***manipulate*** *it. By using computer processing power, we don't need to type in every single command as written text, it should be possible to imagine a system in which the interaction is as intuitive as possible, while interacting with objects that are not, at first, intuitive at all (e.g. the Catmull spline).*

As visual appendices, displays are also means to expand the range of our interactions. While the first means of interaction was purely mechanically written (i.e. typographical), having a spatial display now allows us to organize information in a non-linear fashion. Similarly to how Sutherland talks about kinesthetic displays, we could potentially think of olfactive displays, or kinetic displays—essentially vibrating smartphones and game controllers.

Slightly tangentially, there is some evidence that a "visual-first approach" is actually somewhat dependent on the fact that Western linguistic cultures are visual cultures, moreso than non-Western linguistic cultures which, such as Farsi or Cantonese, have a richer vocabulary when it comes to tasting things. Additionally, visual cues are not as good as olfactive cues (smells) when it comes to helping memory.

Displays allow us to:

- *see-through matter: seeing two things at the same time*
- *involve all human senses*

- - *scent as an archive, like a journal/spotify*
- - *smell as mood: meditation app with different smells*
- - *ordering ingredients from the markets (smell and taste)*
- - *smell as accessibility: trash*
- - *smell as work mood*

—

comparing and **relating** are two ways to create knowledge.

computers make it possible to compare the invisible world with the visible world.

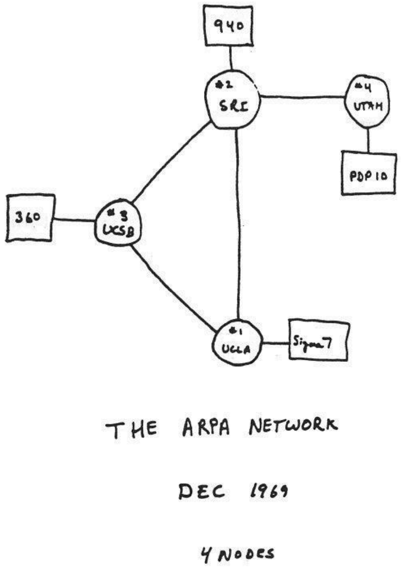
- the preservation of favored traces , reveals the hidden evolution of a text through the comparison of the different editions.
- growing regions , compares the different approaches to the built environemtn, and their relationships.

—

Dreams of networks

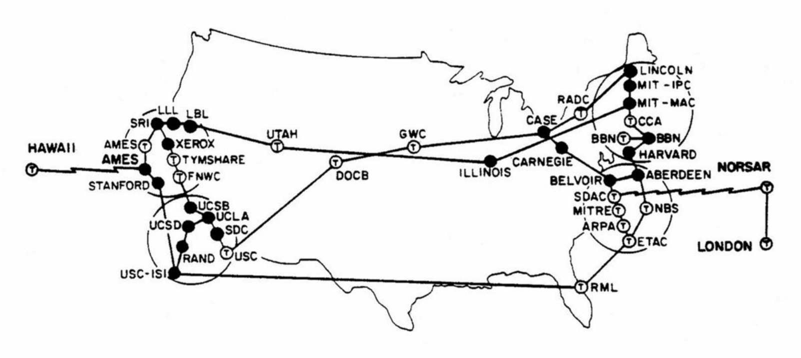
–

The internet is a bunch of computers connected by cables.



The ARPANet in 1969

–



The ARPANet in 1973

–

The Internet can carry a lot of different kinds of information

(like the telegraph carries single words, or the phone carries soundwaves)

–

Web 1.0

–

Tim Berners-Lee is _the_ web developer

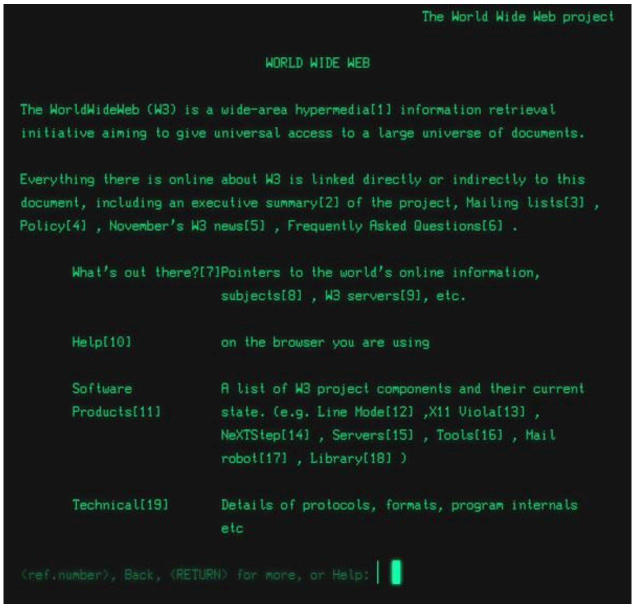
The Web was developed at CERN in order to exchange *documents*, that were formatted in a language called *HTML*.

In the 1980s, Tim Berners-Lee came up with a way to exchange documents across the Internet. These documents became webpages, made up of hypertext, connected by hyperlinks and viewed via web browsers.

The keyword is hyper:

- *hypertext markup language (html)*
- *hypertext transfer protocol (http)*
- *hyperreference link*

–

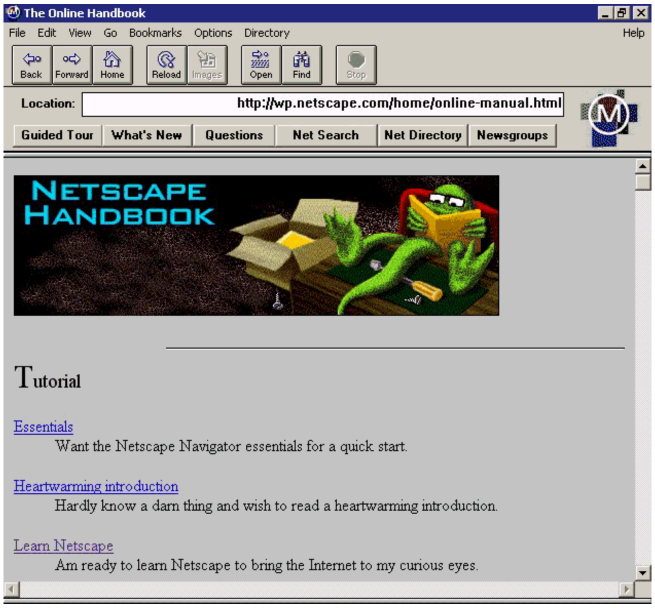


```
                                                  The World Wide Web project

                                WORLD WIDE WEB

The WorldWideWeb (W3) is a wide-area hypermedia[1] information retrieval
initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this
document, including an executive summary[2] of the project, Mailing lists[3] ,
Policy[4] , November's W3 news[5] , Frequently Asked Questions[6] .

       What's out there?[7]Pointers to the world's online information,
                           subjects[8] , W3 servers[9], etc.

       Help[10]            on the browser you are using

       Software            A list of W3 project components and their current
       Products[11]        state. (e.g. Line Mode[12] ,X11 Viola[13] ,
                           NeXTStep[14] , Servers[15] , Tools[16] , Mail
                           robot[17] , Library[18] )

       Technical[19]       Details of protocols, formats, program internals
                           etc

<ref.number>, Back, <RETURN> for more, or Help:
```

The first webpage

That webpage was essentially describing itself.

—



Location: http://wp.netscape.com/home/online-manual.html

Tutorial

Essentials
Want the Netscape Navigator essentials for a quick start.

Heartwarming introduction
Hardly know a darn thing and wish to read a heartwarming introduction.

Learn Netscape
Am ready to learn Netscape to bring the Internet to my curious eyes.

The first popular browser

the earliest and most popular of which was Netscape Navigator.

Netscape was a browser that allowed you to display images and click on links, which made the web a lot more accessible for the broader audience. It also introduce the JavaScript language.

–

Web 2.0

–

The web 2.0 refers to when the WWW becomes a lot more usable

- lowering the barrier to entry
- interacting on with websites
- the *social* web and the *mobile* web

–

what is at stake:

- democratization vs. corporatization
- information vs. attention

–

Structure and style

–

HTML review

–

The *HyperText Markup Language* describes what makes up a webpage.

(i.e. a document meant to be served over the web protocol via web browsers)

–

An HTML element is made up of:

- opening and closing tags
- contents
- attributes

```
<a href="https://webpage.com"> click here </a>
```

–

Examples of tags:

- **p** for paragraph
- **h1** for headings
- **a** for links
- **img** for images
- **ul**, **ol** for lists
- **li** for list items
- ...

–

CSS basics

–

A CSS file contains all of the style rules to be applied to a document.

CSS makes form separate from content .

—

A style rule targets elements and sets the value of attributes.

```
element {
    attribute: value;
}
```

```
a {
    color: red;
    font-size: 14px;
}
```

—

We need to tell the HTML which CSS rules to apply by **linking** the CSS documents:

```
<link rel="stylesheet" href="style.css" />
```

—

Layout

—

Layout is done by putting things in boxes and manipulating the size and the space between those boxes.

An HTML box is called a **div**.

We can manipulate **width**, **height**, **margin**, **padding**.

—

Introducing **flexbox**.

```
container {
    display: flex; /* this line unlocks flexbox layout for the
children of the container */
    flex-direction: row; /* this tells the children to align as a
row (the default is column) */
}
```

(full guide here)

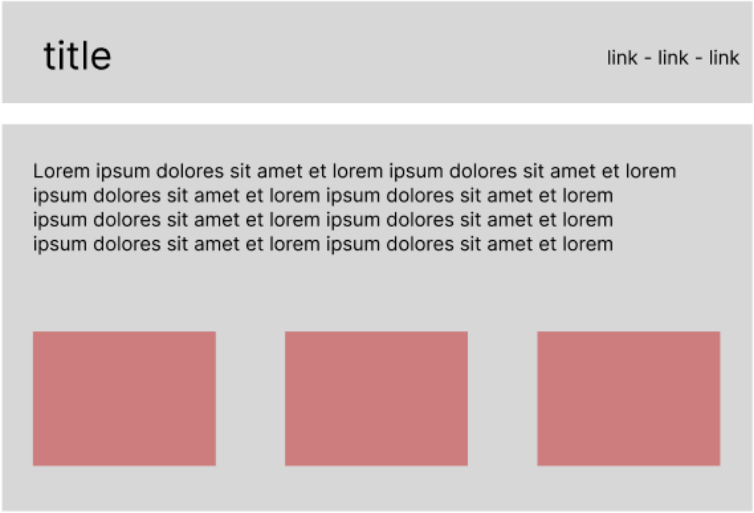
–

Exercise

–

On a separate page, make a header with a title on the left and some links on the right.

Inside the main part of the webpage, put some text, and then a row of three images.



Lorem ipsum dolores sit amet et lorem ipsum dolores sit amet et lorem ipsum dolores sit amet et lorem ipsum dolores sit amet et lorem ipsum dolores sit amet et lorem ipsum dolores sit amet et lorem ipsum dolores sit amet et lorem ipsum dolores sit amet et lorem

Recreate this layout

–

Media queries

—

Media queries is a system allowing you to **change the style depending on the screen size**.

```
@media only screen and (min-width: 600px) {

/* here, we write any rule that will only be applied if our screen is BIGGER THAN 600PX */

}
```

This is very useful when you want to adapt things to a smartphone screen, for instance! This modular approach to webdesign is called "media queries".

One thing we often change is width of the main container (content on a phone usually takes the whole width, but not on laptops).

You can read about it more on w3schools .

—

Outro

—

The dreams of computers were dreams of helping us think, which requires an act of **translation** (into machine languages) and **representation** (via screen interfaces).

—

On a webpage, everything is a box. By using whitespace and flexbox on those boxes, we can do whatever layout we want!

Homework:

- Write your reading response on a new page on your website. Add some images, and use flexbox to lay them out nicely. Add some margin and padding to organize white space. Make sure the layout is responsive on mobile, using media-query (see how to view the mobile version on chrome)

—

Appendix