# Laws of UX

A Note By
Shah Md. Arshad Rahman Ziban

## 1. Jakob's Law

Users spend most of their time on other sites, and they prefer your site to work the same way as all the other sites they already know.

**Jakob's Law (Jakob Nielsen, 2000)**

### Key Takeaways:

- Users expect familiarity - They transfer their expectations from familiar websites or apps to new ones.
- Match mental models - Design interfaces that align with what users already know to reduce cognitive load.
- Minimize unnecessary friction - Use common design patterns (e.g., navigation, layout, search placement) to help users complete tasks easily.
- Support transitions - When changing designs, allow users to use the old version temporarily to avoid frustration.

Jakob's Law doesn't mean all websites should look the same—it encourages starting with familiar design patterns to help users feel comfortable and confident from the start.

## 2. Fitts's Law

The time to acquire a target is a function of the distance to and size of the target.

**Fitts's Law (Paul M. Fitts, 1954)**

The faster and more accurately a user can interact with a UI element depends on two things:
1. How far it is from their current pointer position.
2. How big the element is.

### Fitts's Law Formula

$$ID = \log2\ (\frac{2D}{W})$$

Where:
ID = Index of Difficulty (in bits)
D = Distance from starting point to the center of the target (pixels or cm)
W = Width of the target (pixels or cm)

### As a UI/UX Designer, What We Need:

We don't always need exact physical measurements — you can use pixel distances in your design tool (like Figma, XD, or Sketch).

**D (Distance):** Measure from the center of the user's focus point (e.g., center of screen, previous button) to the center of the target.

**W (Width):** Width or height of the clickable object (whichever direction you're measuring along).

### Example in UI Design

Let's say we have a "Submit" button:
It is 200 pixels away from the form field the user just filled. The button is 100 pixels wide.

### Now Calculate:

$$ID = \log_2\left(\frac{2 \times 200}{100}\right) = \log_2(4) = 2 \text{ bits}$$

Lower ID = faster + easier interaction.

## 3. Hick's Law

**Formulated by:** William Edmund Hick and Ray Hyman in 1952.

**Purpose:** To study how the number of choices affects reaction time.

**Finding:** As the number of options increases, decision time increases logarithmically.

### Formula:

$$RT = a + b \log_2(n)$$

RT = Response Time
n = Number of options
a, b = Constants depending on the task

### Example Scenario:

Suppose in a user interface (UI), a user is presented with 8 buttons to choose from.

**Let's assume:**

a=0.5 seconds (base reaction time)

b=0.2 seconds (scaling factor based on task difficulty)

n=8 (number of choices)

### Calculation:

$$RT = 0.5+0.2 \log 2(8)$$

$$RT = 1.1 \text{ seconds}$$

The user will take approximately 1.1 seconds to make a decision when presented with 8 options, assuming the given constants.

## 4. Miller's Law

Miller's Law comes from a study by psychologist George A. Miller in 1956. He found something interesting about how much information people can hold in their short-term memory.

Most people can remember about 7 things at once — sometimes a little more (9) or a little less (5). That's why it's often called:

**The Magical Number 7, Plus or Minus 2.**

### What is "Chunking"?

Miller noticed that we don't always remember individual pieces, but we group things into chunks.

### Example:

Hard to remember: 9 1 7 4 6 2 5 3 8 0

Easier to remember: 917-462-5380

This is chunking — grouping bits into bigger, familiar parts so we can remember them better.

### How This Helps in UX Design

In design (especially in apps, websites, or forms), you should:

**Keep Choices Limited**
Don't show too many buttons or options. Aim for around 5 to 9.

**Group Related Info Together**
Use headings, sections, and spacing to organize things into meaningful parts.

**Make Content Easy to Scan**
Break long text into short paragraphs, bullets, or steps.

## 5. Postel's Law

Postel's Law suggests a very helpful rule in UX (User Experience) and interface design:

> **Be strict in what you ask the user to do, but flexible in what you accept from the user.**