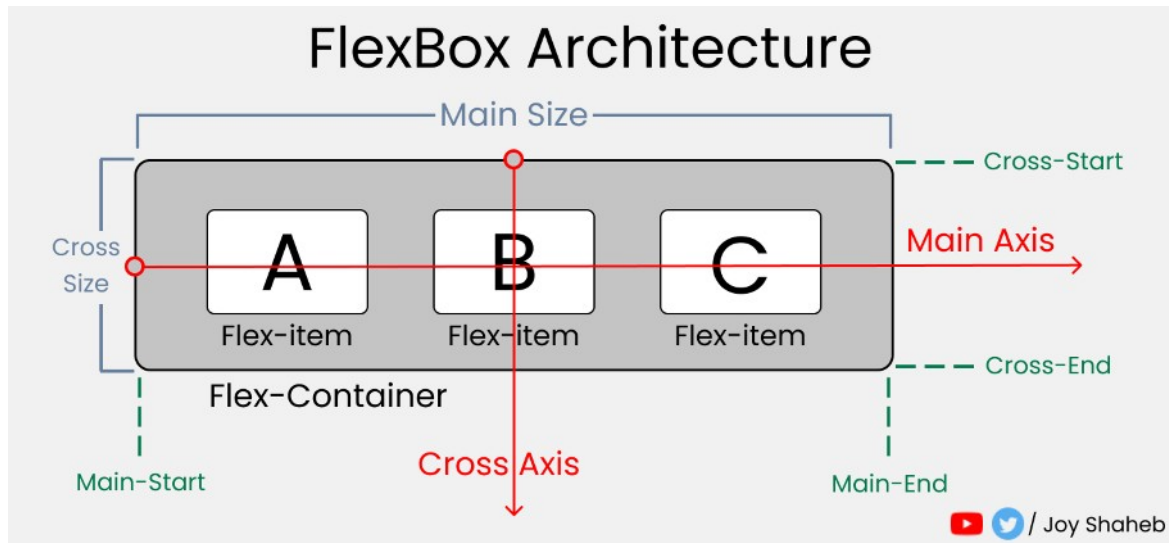


# Заняття 21.

## 1. Повторення



### Property

- #1 display
- #2 flex-direction
- #3 justify-content
- #4 align-items
- #5 align-content
- #6 align-self
- #7 Order
- #8 flex-grow
- #9 flex-shrink
- #10 flex-wrap

### Value(s)

flex  
row || column || column-reverse || row-reverse  
flex-start || flex-end || center ||  
space-between || space-around ||  
space-evenly  
flex-start || flex-end || center ||  
stretch || baseline  
flex-start || flex-end || center ||  
stretch || space-between || space-around  
auto || flex-start || flex-end || center ||  
baseline || stretch  
/\* Any positive Value \*/  
/\* Any positive Value \*/  
/\* Any positive Value \*/  
nowrap || wrap || wrap-reverse

flex-direction :

row



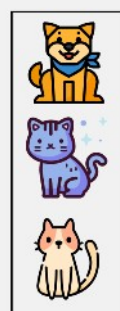
row-reverse

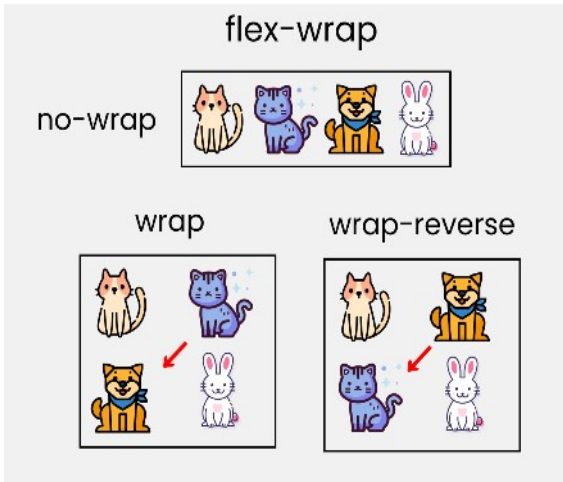
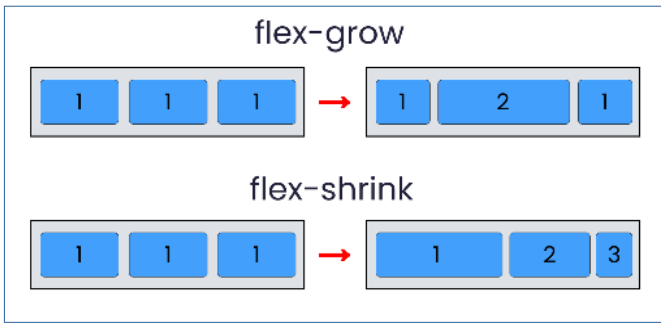
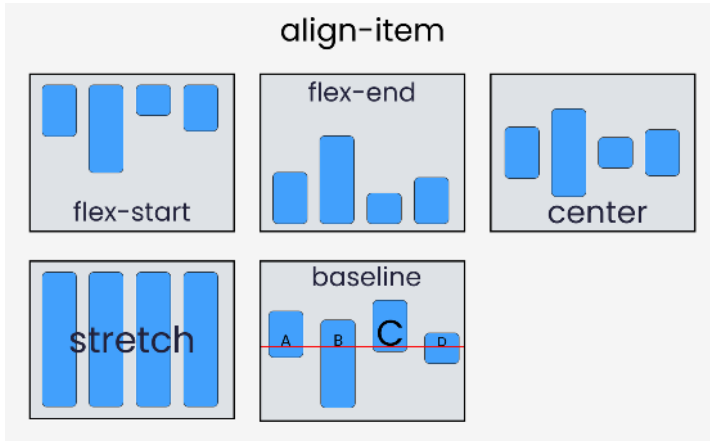
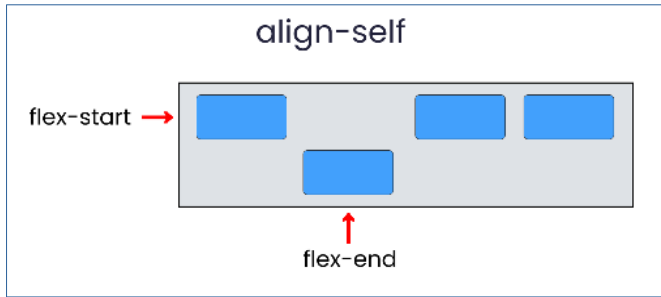
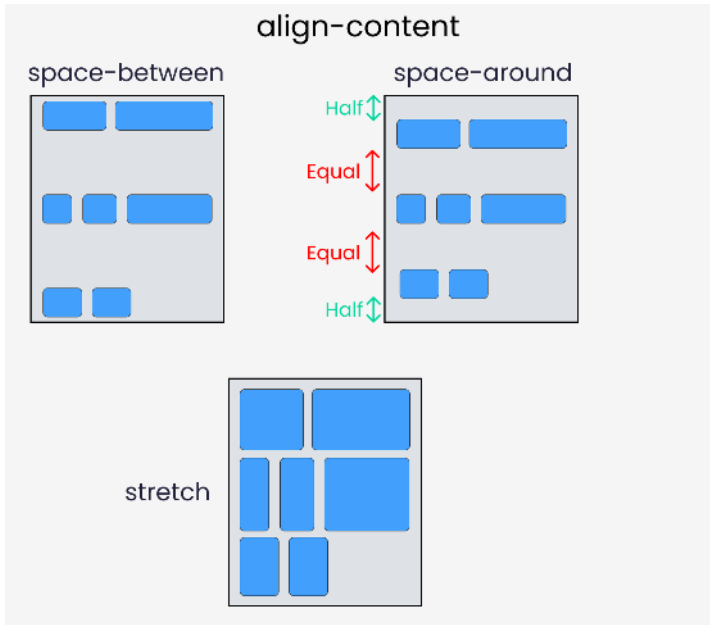
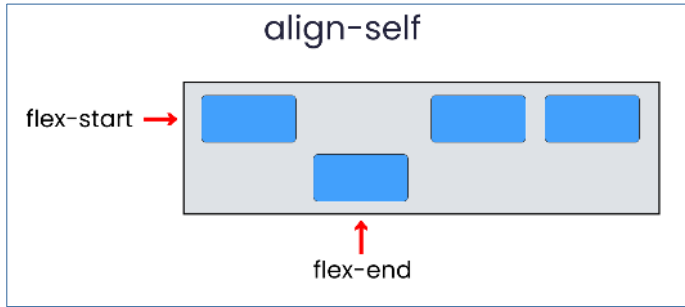
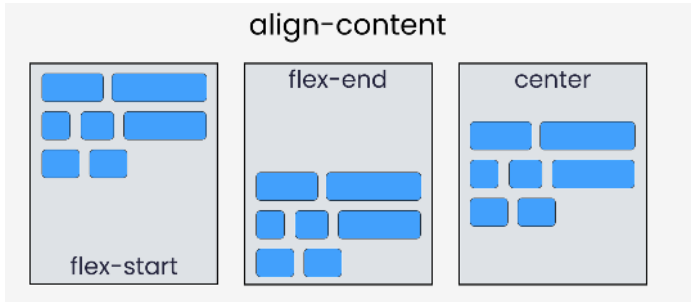
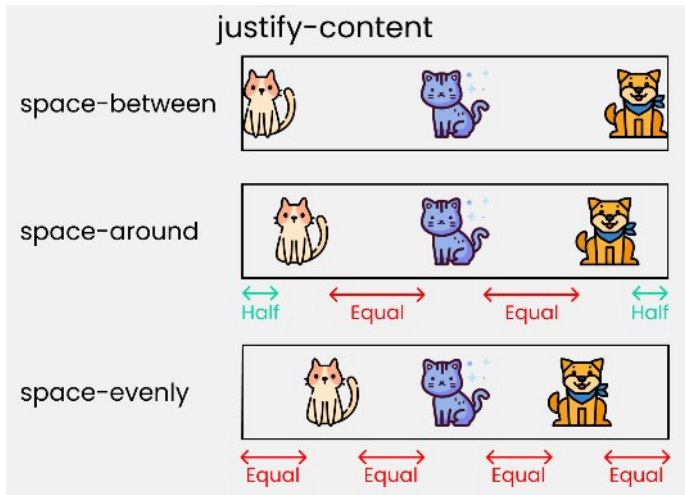
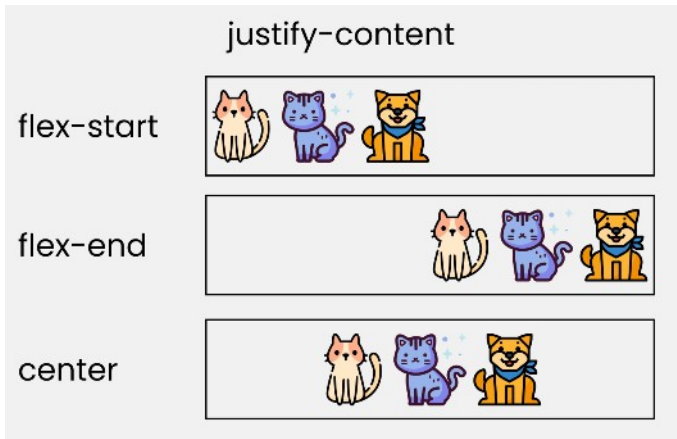


column

column-reverse

flex-direction





## 2. Анімація



Правило **@keyframes** встановлює ключові кадри при анімації елемента.

```
@keyframes box {  
  from {left: 0;}  
  to {left: 300px;}  
}
```

Ключовий кадр - це властивості елемента (прозорість, колір, положення та ін.), які повинні застосовуватися до елемента в заданий момент часу. Таким чином, анімація представляє собою плавний перехід стилізових властивостей від одного ключового кадру до іншого.

Обчислення проміжних значень між такими кадрами бере на себе браузер.

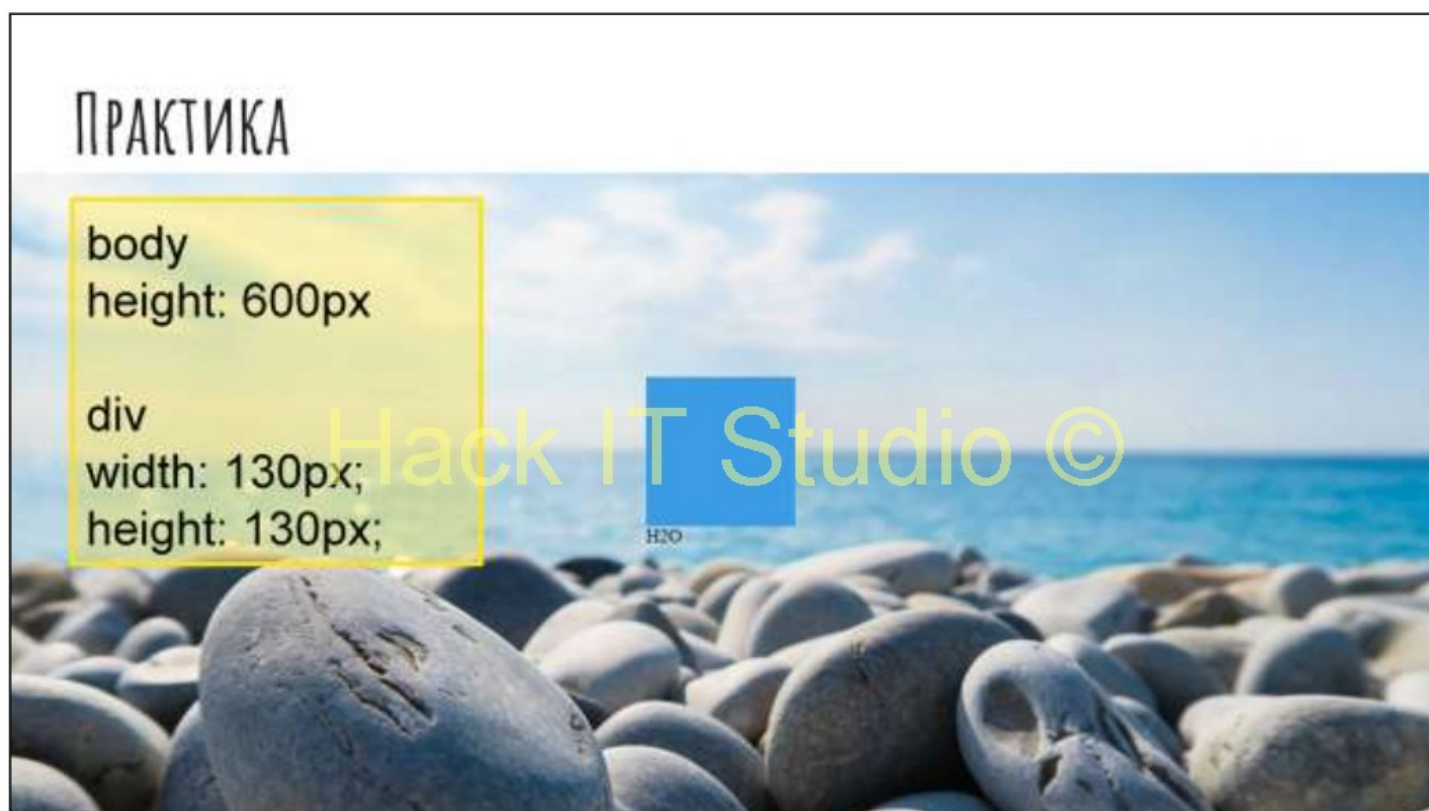
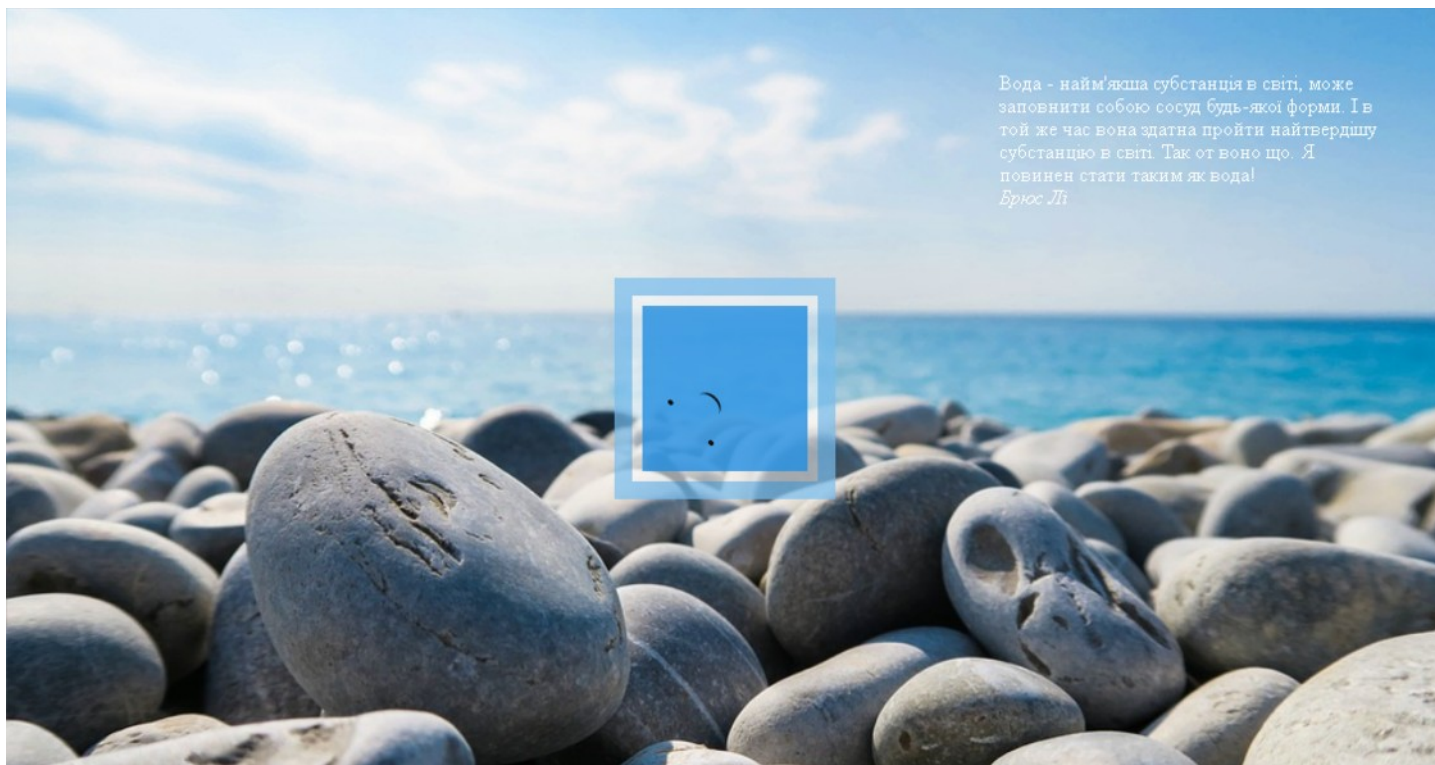
Для створення анімації, **@keyframes** використовується разом із однієї або декількома властивостями для керування анімаціями, або універсальною властивістю **animation**. **@keyframes** дозволяє встановлювати стилі елемента у різні періоди анімації. Якщо точніше, це правило визначає стилі елемента в різні моменти ітерації (циклу) анімації. Ви можете використовувати так властивості як:

- **animation-name** - Встановлює одну або кілька анімацій, які застосовуються до елемента. Являє собою ім'я, пов'язане з правилом **@keyframes**, воно в свою чергу задає послідовність руху.
- **animation-duration** - задає час в секундах або мілісекундах, скільки повинен тривати один цикл анімації. За замовчуванням значення дорівнює 0s, це означає, що ніякої анімації немає.
- **animation-timing-function** - встановлює, згідно якої функції часу повинна відбуватися анімація кожного циклу між ключовими кадрами.
- **animation-delay** - встановлює час очікування перед запуском циклу анімації.
- **animation-iteration-count** - властивість визначає, скільки разів програвати цикл анімації до її зупинки.
- **animation-direction** - встановлює напрямок руху анімації.
- **animation-fill-mode** - визначає, які стилі повинні застосовуватися до елемента, коли анімація не програвється.
- **animation-play-state** - властивість визначає, програвати анімацію або поставити її на паузу. або універсальну властивість **animation**, щоб задати швидкість анімації, кількість циклів тощо.



### 3. Завдання 1.

1. Відкрийте файл `index.html` і `style.css` які знаходяться в папці `watter`.



## ПРАКТИКА



```
<div class="dribble">
  <div class="face">
    <div
      class="eyes">

      <div></div>
      <div></div>
    </div>
    <div
      class="mouth">
    </div>
  </div>
```

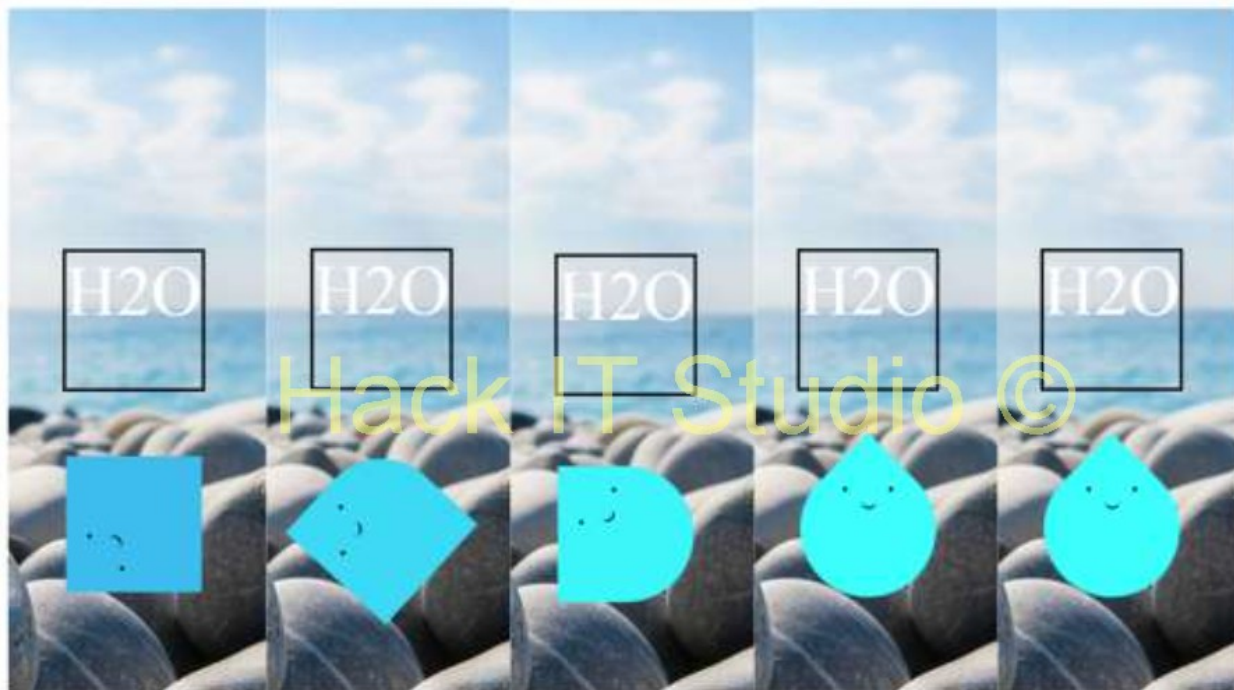
## ПРАКТИКА



```
@keyframes dribble_animate{
  0%{
    bottom: 0px;
    border-radius: 0
    0 0 0;

    transform:
      translate(0, 0)
      scale(1)
      rotate(0deg);
  }
  100%{
    bottom: 0px;
  }
}
```





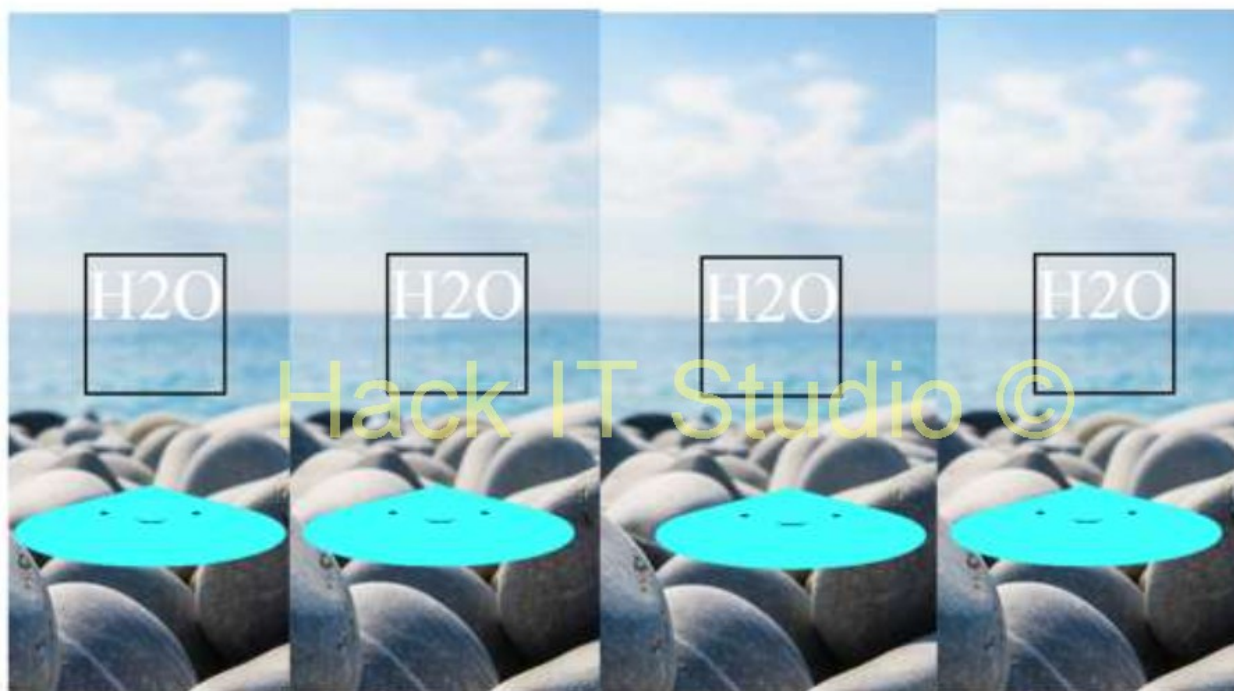
5%

10%

15%

20%

22%

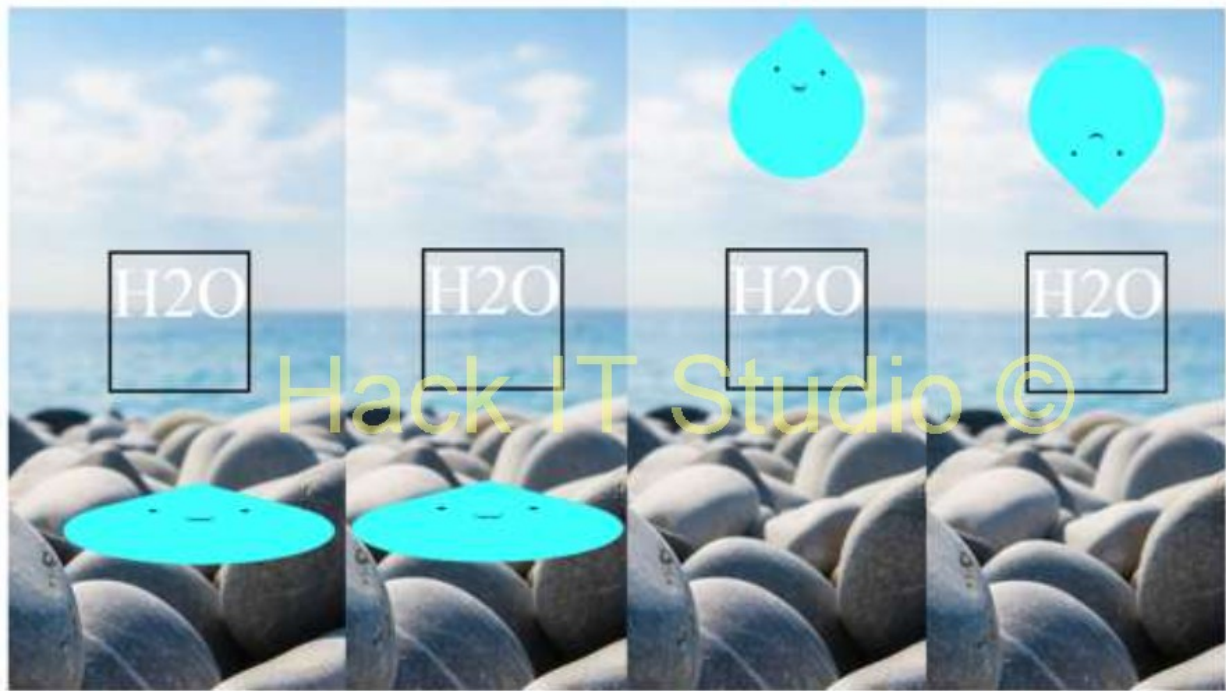


24%

26%

28%

30%

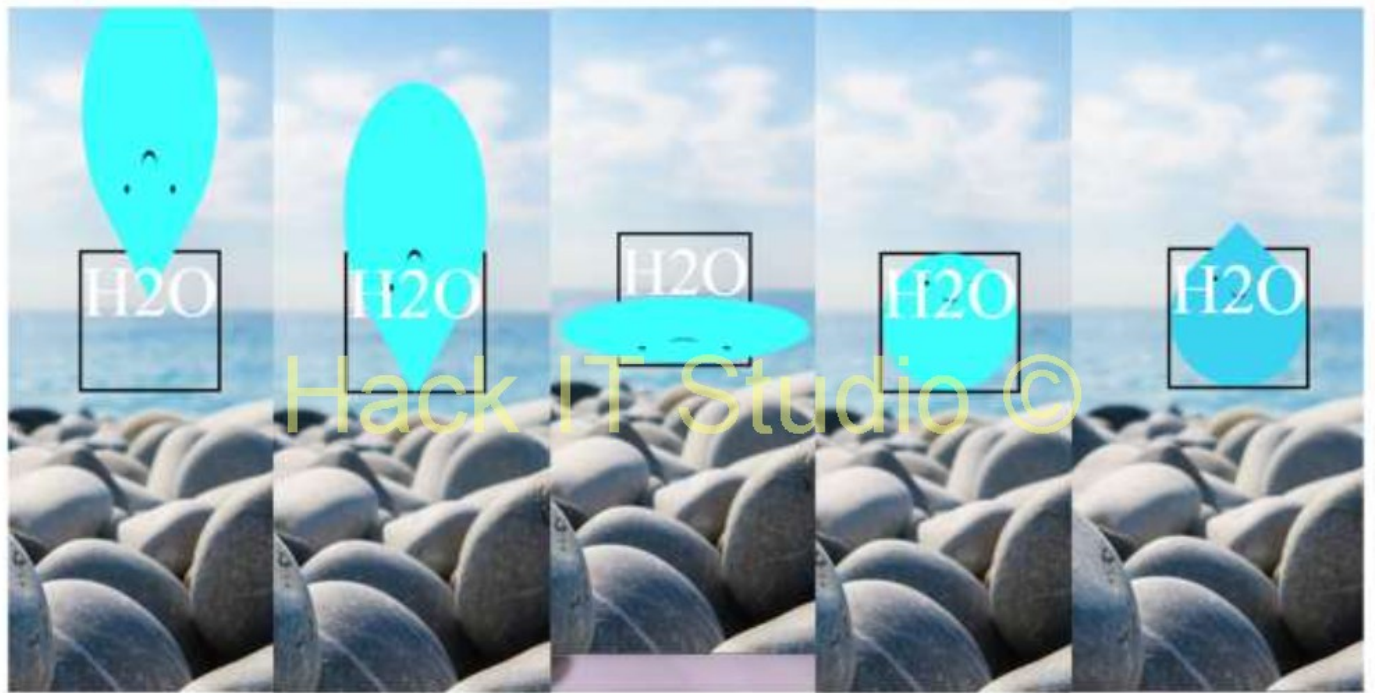


32%

33%

36%

50%



60%

65%

75%

80%

85%, 95%