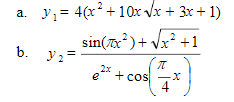
**Session 2 + 3 Biến và kiểu dữ liệu**

Bài luyện tập:

1. Nhập vào bán kính hình tròn, in ra chu vi và diện tích hình tròn.
2. Nhập vào chiều dài và chiều rộng, in ra chu vi và diện tích hình chữ nhật
3. Tính giá trị biểu thức:



với x nhập vào từ bàn phím (gợi ý: tìm hiểu cách sử dụng Math của javascript)

1. Nhập vào số tiền X vnđ (giả sử số tiền là bội số của 10.000vnđ), đổi xem số

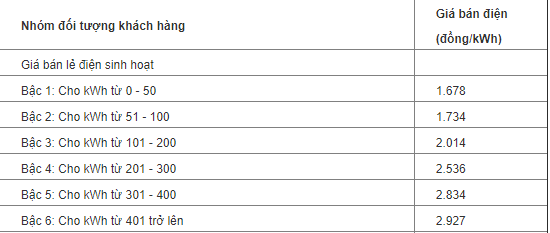
tiền đó được bao nhiêu tờ 100.000, 50.000, 20.000, 10.000 vnđ.

Ví dụ: 320.000 vnđ = 3 tờ 100.000 + 0 tờ 50.000 + 1 tờ 20.000 + 0 tờ 10.000 vnđ

1. Nhập 1 số có 3 chữ số, tính tổng các chữ số của số đó

Ví dụ: số 123 có tổng các chữ số là: 1 + 2 + 3 = 6.

**Session 4 Điều kiện**

1. Nhập vào 1 năm, kiểm tra xem năm đó phải năm đó có bao nhiêu ngày?
2. Nhập vào 1 tháng trong năm, in ra màn hình số ngày của tháng đó (Yêu cầu kiểm tra cả năm nhuận).
3. Nhập vào 1 tháng trong năm, in ra mùa của tháng đó
4. Nhập vào 3 số a, b, c. In ra theo thứ tự tăng dần.
5. Nhập vào 3 số a, b, c. Kiểm tra xem 3 số đó có lập được thành tam giác không. Nếu có, kiểm tra xem tam giác có vuông, cân hay đều k?
6. Cho thông tin bảng giá điện sau. Nhập vào số điện sử dụng của 1 hộ trong tháng. In ra chi tiết số tiền từng bậc và tổng số tiền phải trả.
7. Cho người dùng nhập vào chiều cao, cân nặng. Tính chỉ số BMI, in ra tình trạng hiện tại của người dùng theo cấp độ của chỉ số BMI.

**Session 5 Vòng lặp**

Bài luyện tập:

1. In ra các số từ 0 đến 100
2. In ra các số từ 100 về 0
3. In ra các số chẵn/lẻ từ 0 đến 100.
4. In ra các số chẵn/lẻ từ 100 về 0.
5. Thực hiện tính tổng: S = 1 + 2 + 3 + … + N với N nhập từ bàn phím.
6. Thực hiện tính tổng: S = 1 + 3 + 5 + … + 2N + 1 với N nhập từ bàn phím.
7. Thực hiện tính tổng: S = 2 + 4 + 6 + … + 2N với N nhập từ bàn phím.
8. Thực hiện tính tổng: S = với N nhập từ bàn phím.
9. Thực hiện tính tổng: S = với N nhập từ bàn phím
10. Cho biết dãy số Fibonacci: 1, 1, 2, 3, 5, 8, …

có dạng: (số sau bằng tổng 2 số liền trước):

- Tìm phần tử thứ N của dãy số với N nhập từ bàn phím.

- Tổng N phần tử đầu tiên của dãy số với N nhập từ bàn phím.

1. In ra màn hình hình sau:  
   \*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*

với chiều dài là a dấu \*, chiều rộng là b dấu \* (a, b nhập từ bàn phím).

1. In ra màn hình hình sau:

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

với chiều dài cạnh dưới cùng là a dấu \* (a nhập từ bàn phím).

1. Yêu cầu người dùng nhập mật khẩu, nếu mật khẩu ngắn hơn 8 ký tự, yêu cầu nhập lại.
2. Yêu cầu người dùng nhập mật khẩu, nếu mật khẩu ngắn hơn 8 ký tự, thông báo người mật khẩu quá ngắn và yêu cầu nhập lại.
3. Yêu cầu người dùng nhập mật khẩu chứa đồng thời 3 ký tự đặc biệt ($, %, \_) và dài ít nhất 8 ký tự. Thiếu điều kiện nào, yêu cầu người dùng nhập lại.
4. Nhập vào 1 số, kiểm tra xem có phải số chính phương không? có phải số nguyên tố không? có phải hợp số k? Nếu là hợp số, in ra các ước số của nó.   
   (chú thích: số chính phương là bình phương của 1 số tự nhiên nào đó, ví dụ: 25 là số chính phương. Số nguyên tố là số chỉ chia hết có 1 và chính nó, ví dụ: số 11. Hợp số là số chia hết cho các số ngoài 1 và chính nó. B được gọi là ước số của A nếu A chia hết cho B).
5. **Chuẩn hóa họ tên. Ví dụ: " nGuyễN văn tuấn " => "Nguyễn Văn Tuấn".**

**(Cắt bỏ các dấu cách ở đầu, cuối,**

**giữa các từ chỉ có 1 dấu cách,**

**chỉ viết hoa chữ cái đầu tiên mỗi từ)**

**Session 6 Mảng**

1. Tính tổng các phần tử của 1 mảng
2. Tính trung bình các phần tử của mảng
3. Tìm số lớn nhất, nhỏ nhất
4. Sắp xếp mảng
5. Tìm số lớn thứ hai, nhỏ thứ hai của mảng
6. Tìm các số trong mảng lớn hơn N nhập từ bàn phím.
7. Cho 1 mảng: [“Javascript”, “Html”, “Css”, “Pascal”], nhập 1 giá trị từ bàn phím, tìm các giá trị trong mảng chứa từ nhập vào.  
   Ví dụ: Nhập “java” => Javascript, nhập “s” => Javascript, Css, Pascal
8. Đảo ngược chuỗi: nhập vào 1 chuỗi, in ra chuỗi bị đảo ngược.  
   ví dụ: “Hello world” => “dlrow olleH”
9. **Cho 2 mảng arr1, arr2. In ra mảng arr3 gồm các phần tử thuộc cả mảng arr1 và arr2.**
10. **Giả sử 1 gia đình nuôi 1 đàn gà (giả sử 10 con). Tạo 1 mảng lưu trọng lượng của đàn gà này.**

**a. Giả sử sau 2 tuần, trong lượng mỗi con trong đàn đều tăng gấp rưỡi. Viết đoạn code thực hiện việc này**

**b. Gia đình muốn tìm con gà có trọng lượng lớn nhất để bán. Tìm con gà này**

**c. Có 1 đợt dịch H5N1 tràn về, đàn gà bị chết 3 con (chọn ngẫu nhiên), và những con còn lại trọng lượng bị giảm 1 nửa. Hãy viết code thực hiện việc này**

1. **Cho 1 mảng số nguyên, kiểm tra xem trong mảng có bao nhiêu phần tử khác nhau, mỗi phần tử xuất hiện bao nhiêu lần**
2. **2 mảng được gọi là bằng nhau nếu toàn bộ phần tử mảng này đều nằm trong mảng kia và ngược lại. Viết code so sánh 2 mảng a và b có bằng nhau hay không?**
3. **Tương tự, mảng a nằm trong mảng b nếu toàn bộ phần tử mảng a đều thuộc mảng b. Viết code thực hiện so sánh này**

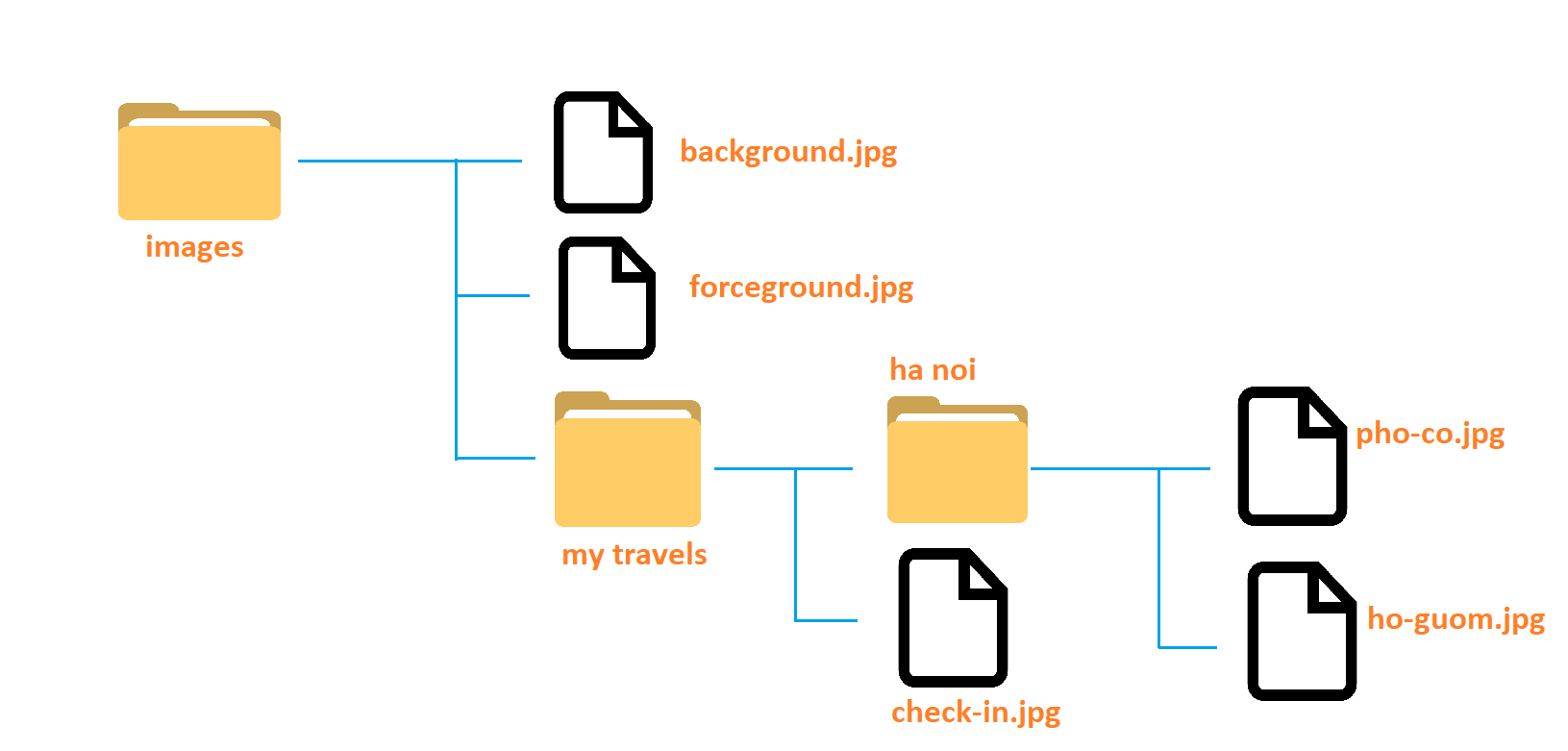
**Session 7 Object**

1. Tạo object chứa các thông tin của person gồm key: name, age, favorite với các giá trị tương ứng: “Phạm Văn A”, 20, [“Javascript”, “Html”, “Css”].
2. Tạo thêm key cho person: “school”, giá trị: “ĐH Quốc Gia HN”, k thay đổi khai báo ban đầu.
3. Tạo thêm key, value, đều nhập từ bàn phím

**Các bài tập luyện tập chung.**

1. Tính các tổng S sau:
   1. S = với x và N nhập từ bàn phím.
   2. S = 1! + 2! + 3! + … + N! với N nhập từ bàn phím.
2. Tìm N nhỏ nhất sao cho:

S = 1 + 2 + 3 + 4 + … + N > 1000

1. Cho 2 số a, b: tìm UCLN(a,b) và BCNN(a,b);
2. Nhập một số nguyên N, in ra cách đọc số nguyên đó.
3. Bạn cần tìm file dựa theo tên trong 1 thư mục, yêu cầu tìm tất cả file thỏa mãn tên do người dùng nhập vào. Mô phỏng cấu trúc dữ liệu như sau:  
   Object mô tả 1 file hay 1 folder:   
   {  
    “name”: “image\_background.jpg”,  
    “isFolder”: true,  
    “content”: []  
   }  
   Giải thích:  
    - name: mô tả tên của file hoặc thư mục - kiểu string  
   - isFolder: mô tả object này là file hay folder - true: là folder, false: là file - kiểu boolean.  
   - content: nếu object là file, content k có nội dung (mảng rỗng []), nếu object là folder, sẽ lưu danh sách các object khác.  
   Ví dụ:  
     
   Cấu trúc dữ liệu tương ứng:  
     
   let images = {   
    "name": "images",   
    "isFolder": true,  
    "content":[  
    {   
    "name": "background.jpg",  
    "isFolder": false,   
    "content": []  
    },  
    {  
    "name": "forceground.jpg",  
    "isFolder": false,  
    "content": []  
    },  
    {  
    "name": "my travel",  
    "isFolder": true,  
    "content": [  
    {  
    "name": "ha noi",  
    "isFolder": true,  
    "content": [  
    {  
    "name": "pho-co.jpg",  
    "isFolder": false,  
    "content": []  
    },  
    {  
    "name": "ho-guom.jpg",  
    "isFolder": false,  
    "content": []  
    }  
    ]  
    },  
    {  
    "name": "check-in.jpg",  
    "isFolder": false,  
    "content": []  
    }  
    ]  
    }  
    ]  
   }  
     
   Yêu cầu: cho người dùng nhập 1 từ khóa, tìm tất cả file có chứa từ khóa đó, k phân biệt chữ hoa thường.
4. Tạo trang web về 1 bài tập trắc nghiệm. Tham khảo câu hỏi và giao diện ở [link](https://timoday.edu.vn/trac-nghiem-javascript/) này  
   Link demo: [link](https://nguyenxuanhung96.github.io/trac-nghiem-js-html-css/)   
   Link source: [link](https://github.com/nguyenxuanhung96/trac-nghiem-js-html-css)
5. Giả sử 1 gia đình nuôi 1 đàn gà (giả sử 10 con). Tạo 1 mảng lưu trọng lượng của đàn gà này.
6. Giả sử sau 2 tuần, trong lượng mỗi con trong đàn đều tăng gấp rưỡi, viết 1 function nhận vào 1 mảng, function thực hiện thay đổi này.
7. Gia đình muốn tìm con gà có trọng lượng lớn nhất để bán. Viết 1 function theo yêu cầu này.
8. Có 1 đợt dịch H5N1 tràn về, đàn gà bị chết 3 con (chọn ngẫu nhiên), và những con còn lại trọng lượng bị giảm 1 nửa. hãy viết function (nhận vào 1 mảng) mô tả đợt dịch này.
9. Game nho nhỏ, Escape, có 3 thực thể, P(player), K(key), E(escape), P sẽ di chuyển trên 1 mê cung WxH (width x height), P chỉ thoát được thoát ra khi đi lấy được K và đi vào E. Code sẽ vận dụng kiến thức từ bài Object đổ về, mn xem thử [video](https://drive.google.com/file/d/1IB4RmKuQI1Gtj-_lCfdZ-_sq8nspT0ZC/view?usp=sharing) cho dễ hình dung.  
   Yêu cầu:

* a. Cho người dùng nhập width, height của map
* b. 3 điểm P, K, E phải random vị trí, ko đc trùng nhau
* c. Không cho đi ra ngoài map
* d. Di chuyển P bằng cách cho người dùng nhập 1 trong 4 phím w, s, a, d
* e. Game chỉ dừng khi đi vào K rồi mới đến E.

**Các bài luyện tập thêm về ngôn ngữ Javascript**

1. Tìm hiểu cách sử dụng Datetime.
   1. In ra thời điểm hiện tại, bao gồm thứ, ngày tháng năm.
   2. Tính số ngày chênh lệch giữa 2 khoảng thời gian.
   3. Chuyển 1 chuỗi có định dạng thành 1 đối tượng Datetime, từ đó lấy được thông tin về ngày giờ phút, thứ ngày tháng năm của chuỗi đó.
2. Tìm hiểu cách sử dụng Math.
   1. Các hằng số được cung cấp sẵn
   2. Các phép toán như lấy mũ, khai căn, giá trị tuyệt đối, làm tròn lên/xuống, sin, cos.
3. Tìm hiểu cách sử dụng Random.
   1. Cách lấy một số tự nhiên ngẫu nhiên từ 0 đến N.
   2. Cách lấy một số tự nhiên ngẫu nhiên từ M đến N

(M, N nhập từ bàn phím)

1. **Tìm hiểu setTimeout và setInterval:  
   Tạo 1 đồng hồ đếm ngược, hiển thị theo format: ? ngày, ? giờ ? phút ? giây.  
   Hết giờ sẽ hiển thị 1 alert thông báo.**
2. Api Ajax: [https://weather-data-demo.herokuapp.com](https://weather-data-demo.herokuapp.com/)