

Quiz 09

Câu 1

- Phân biệt hàm `ceiling`, `floor` với `round`
- Trường hợp nào các hàm trên có giá trị giống nhau ?
- Trường hợp nào các hàm trên có giá trị khác nhau ?

Câu trả lời

- Phân biệt:
 - `ceiling`:
 - Đối số truyền vào là số nguyên: Trả về giá trị bằng chính đối số truyền vào. VD: `ceiling(2) = 2`
 - Đối số truyền vào là số thực: Trả về số nguyên nhỏ nhất mà lớn hơn đối số truyền vào. VD: `ceiling(2.4) = 3`
 - `floor`:
 - Đối số truyền vào là số nguyên: Trả về giá trị bằng chính đối số truyền vào. VD: `floor(2) = 2`
 - Đối số truyền vào là số thực: Trả về số nguyên lớn nhất mà nhỏ hơn đối số truyền vào. VD: `floor(2.4) = 2`
 - `round`:
 - Đối số truyền vào là số nguyên: Trả về giá trị bằng chính đối số truyền vào. VD: `round(2) = 2`
 - Đối số truyền vào là số thực: Trả về giá trị làm tròn tới số nguyên gần nhất. VD: `round(2.5) = 3`, `round(2.4999) = 2`
- Trường hợp đối số truyền vào là một số nguyên các hàm trên có giá trị trả về giống nhau
- Trường hợp đối số truyền vào là số thực giá trị của 2 hàm `ceiling` và `floor` thì khác nhau, còn giá trị trả về của hàm `round` có thể bằng 1 trong 2 function trên

Câu 2

- Phân biệt hàm `starts-with` và hàm `contains`

Câu trả lời:

- `starts-with(str1, str2)`: Trả về true nếu str1 bắt đầu bởi str2, ngược lại thì trả về false. VD: `starts-with('XML','X') = true`, `starts-with('XML','A') = false`
- `contains(str1, str2)`: Trả về true nếu str1 có chứa str2, ngược lại thì trả về false. VD: `starts-with('XML','ML') = true`, `starts-with('XML','A') = false`

Câu 3

Nêu kết quả các hàm sau

Câu trả lời:

- `translate("bar","abc","ABC")`: BAr
- `translate("Nhu co bac ho trong ngay vui dai thang","abcdefghijklmnopqrstuvwxyz","ABCDEFGHIJKLMNOPQRSTUVWXYZ")`: NHU CO BAC HO TRONG NGAY VUI DAI THANG
- `translate("Hom nay la ngay cuoi tuan","atjbefugibmg","GINGUGSBJKSs")`: HoS nGy IG nBGy cSoJ ISGn

Câu 4

Giải thích ý nghĩa của XPath sau

Câu trả lời

- `//*`: Lấy tất cả các thẻ có trong tài liệu
- `//room`: Lấy tất cả các thẻ `room` nằm bất cứ đâu trong tài liệu
- `/renderpass//room`: Lấy tất cả các thẻ `room` nằm bên trong thẻ root `renderpass`
- `/renderpass//room/chair`: Lấy tất cả thẻ `<chair>` là con của thẻ `room`, mà thẻ `room` nằm bất cứ đâu miễn là bên trong thẻ root `renderpass`
- `/renderpass/street/house/room`: Lấy tất cả thẻ `room` là con của thẻ `house`, mà thẻ `house` là con của thẻ `street`, mà thẻ `street` là con của thẻ root `renderpass`
- `/renderpass/*/house/room`: Lấy tất cả thẻ `room` là con của thẻ `house`, mà thẻ `house` là cháu của thẻ root `renderpass`
- `room/chair`: Lấy tất cả thẻ `chair` là con của thẻ root `room`
- `//room/*`: Lấy tất cả thẻ nằm bên trong thẻ `room`, mà thẻ `room` nằm bất cứ đâu trong tài liệu
- `//*[substring(name(),6) mod 2 = 1 and parent::*[name() = 'room']]`: Lấy tất cả các thẻ có chiều dài tên tính từ vị trí thứ 6 đến hết là một số lẻ và thẻ này có thẻ anh em có tên là `room`
- `//*[starts-with(name(),'paperclip')]`: Lấy tất cả các thẻ có tên bắt đầu bởi `paperclip`
- `//*[@sets='wood']`: Lấy tất cả các thẻ có thuộc tính `sets` mang giá trị là `wood`,
- `//*[starts-with(name(),'pencil_box')]`: Lấy tất cả các thẻ có tên bắt đầu bởi `pencil_box`
- `//desk_1 | //desk_2`: Lấy tất cả thẻ `desk_1` và `desk_2` nằm bất cứ đâu trong tài liệu
- `//*[not(contains(name(),'desk_1')) and not(contains(name(),'desk_2')) and not(contains(name(),'blotter_1')) and not(contains(name(),'paperclip_1')) and contains(@sets,'shiny') and not contains(@sets,'burnt')]`: Lấy tất cả thẻ có tên không chứa các giá trị sau ("`desk_1`", "`desk_2`", "`blotter_1`", "`paperclip_1`") và thuộc tính `sets` của thẻ đó phải chứa `shiny`, và thuộc tính `sets` của thẻ đó không chứa `burnt`
- `//*[not(contains(name(),'desk_1')) and not(contains(name(),'desk_2')) and not(contains(name(),'blotter_1')) and not(contains(name(),'paperclip_1'))]`: Lấy tất cả thẻ có tên không chứa các giá trị sau ("`desk_1`", "`desk_2`", "`blotter_1`", "`paperclip_1`")