**You said:**

Concepts of internationalization (i18n) and localization (l10n) i want explanation script in gujarati

**ChatGPT said:**

અંતરરાષ્ટ્રીયકરણ (Internationalization - i18n) અને સ્થાનિકીકરણ (Localization - l10n) એ બે મહત્વપૂર્ણ ટર્મ્સ છે જેનો ઉપયોગ સોફ્ટવેર ડેવલપમેન્ટ અને વેબ ડિઝાઇનિંગમાં થાય છે. આ બંને ટર્મ્સનું સંબંધ વિવિધ ભાષાઓ અને સંસ્કૃતિઓ માટે સોફ્ટવેર અને વેબસાઇટ્સને ઢાલવાની પ્રોસેસ સાથે છે. આ ટર્મ્સનો અર્થ અને વપરાશ નીચે આપેલા છે:

**1. આંતરરાષ્ટ્રીયકરણ (i18n) - Internationalization:**

અંતરરાષ્ટ્રીયકરણ એ એ પ્રક્રિયા છે, જેના દ્વારા કોઈપણ સોફ્ટવેર અથવા વેબસાઇટને વિવિધ ભાષાઓ અને સંસ્કૃતિઓ માટે સરળતાથી ઉપલબ્ધ બનાવવાનો પ્રયાસ કરવામાં આવે છે. આમાં એ ખાતરી કરવાનો સમાવેશ થાય છે કે સોફ્ટવેર પ્રોડક્ટ એ વૈશ્વિક બજારો માટે તૈયાર છે.

અંતરરાષ્ટ્રીયકરણની મુખ્ય વાતો:

* **કોડમાં ભાષાઓ માટે સપોર્ટ ઉમેરવો**: જેનાથી પીઠભૂમિ કોડ ભાષા અને વપરાશકર્તા ઈન્ટરફેસ (UI) ભાગ અલગ અલગ હોઈ શકે.
* **વિશ્વસનીય સમય અને તારીખની અસર**: તારીખ અને સમયનું પ્રદર્શન અલગ અલગ દેશોમાં અલગ હોઈ શકે છે.
* **મુદ્રા (Currency)**: વિવિધ દેશોમાં વિવિધ મুদ্রા યુનિકોડ (Unicode) ના માનક સપોર્ટથી થવા જોઈએ.
* **ફોર્મેટિંગ અને નમૂનાઓ**: મૂળભૂત રકમ, તારીખ અને નંબર ફોર્મેટને ગ

**You said:**

Concepts of internationalization (i18n) and localization (l10n) i want explanation script in gujarati for flutter

**ChatGPT said:**

ફ્લટર (Flutter) એ ગૂગલ દ્વારા વિકસાવવામાં આવેલ એક શક્તિશાળી ક્રોસ પ્લેટફોર્મ ફ્રેમવર્ક છે જે iOS અને એન્ડ્રોઇડ બંને માટે મોબાઇલ એપ્લિકેશન્સ ડેવલપ કરવા માટે વપરાય છે. જ્યારે તમે વૈશ્વિક (International) અને સ્થાનિક (Localized) એપ્લિકેશન ડેવલપ કરો છો, ત્યારે **આંતરરાષ્ટ્રીયકરણ (i18n)** અને **સ્થાનિકીકરણ (l10n)** ખૂબ જ મહત્વપૂર્ણ પરિપ્રેક્ષ્ય બની જાય છે. આ બંને કોન્સેપ્ટ ફ્લટરમાં કેવી રીતે લાગુ પડે છે, તેનો સમજૂતિ નીચે આપવામાં આવી છે.

**1. આંતરરાષ્ટ્રીયકરણ (i18n) - Internationalization in Flutter**

આંતરરાષ્ટ્રીયકરણનો અર્થ એ છે કે તમે તમારી એપ્લિકેશનના કોર લોજિકને આવું ડિઝાઇન કરો કે તે વિવિધ ભાષાઓ, સમય ઝોન, કરન્સી અને અન્ય સ્થાનિકિખણ પરિપ્રેક્ષ્યોને સરળતાથી સપોર્ટ કરી શકે.

ફ્લટર માટે i18nનું કાર્ય કરવા માટે, તમે નીચેના પગલાં લેશો:

* **લોગિકલ સેટઅપ**: તમે તમારી એપ્લિકેશનમાં "intl" પેકેજનો ઉપયોગ કરીને આંતરરાષ્ટ્રીયકરણ સેટ કરશો. આ પેકેજ સાથે તમે ભાષાઓ, ડેટા ફોર્મેટ અને અલગ-અલગ નેવિગેશન માટે કોડ સેટ કરી શકો છો.
* **ઝાવા સ્ક્રિપ્ટ અથવા JSON ફાઇલનો ઉપયોગ**: તમારે JSON ફાઇલો અથવા .arb (Application Resource Bundle) ફાઇલોનો ઉપયોગ કરીને વિવિધ ભાષાઓ માટે મેસેજ અને ટેક્સ્ટ સ્ટોર કરવાના છે. આ રીતે, એપ્લિકેશન આપમેળે કોઈપણ ભાષા અથવા સંસ્કૃતિ માટે રૂપરેખિત થઈ શકે છે.

**હેતુ**: i18n એ એ સુનિશ્ચિત કરે છે કે તમારી એપ્લિકેશન બધું કસ્ટમાઇઝ અને ગ્લોબલ અને સ્થાનિક બજારો માટે તૈયાર છે.

**2. સ્થાનિકીકરણ (l10n) - Localization in Flutter**

સ્થાનિકીકરણ એ એ પ્રક્રિયા છે, જે અંતરરાષ્ટ્રીયકરણ પછી થાય છે, જ્યાં એપ્લિકેશનને ચોક્કસ ભાષા અથવા સંસ્કૃતિ માટે પ્રત્યક્ષ રીતે અનુરૂપ બનાવવું છે.

ફ્લટર માટે l10n કાર્ય કરવું એટલે અલગ-અલગ ભાષાઓ, લોકલ કોન્વેંશન્સ, તારીખ અને સમય ફોર્મેટ, અને વપરાશકર્તા ઇન્ટરફેસ (UI) ભાષાઓના આધારે કસ્ટમાઇઝેશન કરવું.

**Flutter l10n માં કરવા માટેનાં મુખ્ય પગલાં**:

1. **flutter\_localizations પેકેજનો ઉપયોગ**: આ પેકેજ ફ્લટર સાથે કામ કરે છે અને વિવિધ ભાષાઓ માટે ટેક્સ્ટ, સમય ફોર્મેટ, મર્યાદા અને કસ્ટમ લોજિક ઉપલબ્ધ કરે છે.
2. **JSON અથવા ARB ફાઇલો સાથે ભાષા ટાઈમિંગ**: ARB (Application Resource Bundle) ફાઇલોમાં તમારે તમારા દરેક ભાષા માટે અનુવાદ કરી સંગ્રહવા છે.
3. **Intl પેકેજ સાથે ભાષા આધારિત મેસેજિંગ**: આ પેકેજથી તમે dynamically શબ્દોનું અનુવાદ અને નમૂનાઓ પ્રદાન કરી શકો છો, જેમ કે તારીખ અને સમયના ફોર્મેટને સ્થાનિક કરવાની ક્ષમતા.

**l10n માટે ઉદાહરણ**:

dart

Copy

import 'package:flutter/material.dart';

import 'package:intl/intl.dart'; // Localization for formatting

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

localizationsDelegates: [

GlobalMaterialLocalizations.delegate,

GlobalWidgetsLocalizations.delegate,

],

supportedLocales: [

Locale('en', 'US'), // English

Locale('hi', 'IN'), // Hindi

Locale('gu', 'IN'), // Gujarati

],

home: MyHomePage(),

);

}

}

class MyHomePage extends StatelessWidget {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Flutter Localization Example'),

),

body: Center(

child: Text(

'Hello World', // This would be translated based on language

),

),

);

}

}

**i18n અને l10n વચ્ચેનો આધાર:**

* **i18n** એ એ સુનિશ્ચિત કરે છે કે તમારી એપ્લિકેશન વૈશ્વિક સ્તરે અપગ્રેડ માટે સજ્જ છે.
* **l10n** એ એ સુનિશ્ચિત કરે છે કે એપ્લિકેશનને ચોક્કસ ભાષાઓ અને સંસ્કૃતિઓ માટે અનુરૂપ બનાવવામાં આવે.

**નમૂના:**

* i18nના ભાગ તરીકે, તમે એવી એપ્લિકેશન બનાવી શકો છો જે વિવિધ ભાષાઓ (જેમ કે અંગ્રેજી, ગુજરાતી, હિન્દી) માટે ટેક્સ્ટ, કરન્સી અને તારીખ ફોર્મેટને સપોર્ટ કરે છે.
* l10nના ભાગ તરીકે, તમે સ્પષ્ટ અનુવાદો પ્રદાન કરશો જે વપરાશકર્તા ઇન્ટરફેસ માટે સંસ્કૃતિવાર રીતે યોગ્ય હોય.

**મકસદ:**

* **i18n** એ અંતર્ગત વાત છે જ્યારે તમે એપ્લિકેશનનું માળખું આ રીતે ગોઠવો કે તે દરેક દેશ અને ભાષા માટે અનુકૂળ બની શકે.
* **l10n** એ તમારા એપ્લિકેશનના અંતિમ અનુવાદ અને કસ્ટમાઇઝેશનને સંકેત કરે છે, જે વેબસાઇટ અથવા એપ્લિકેશન માટે ચોક્કસ ભાષાઓમાં સામગ્રી રજૂ કરે છે.

આ રીતે, **i18n** અને **l10n** બંનેનો ઉપયોગ કરીને તમે તમારી એપ્લિકેશનને વૈશ્વિક બજારમાં ઉપલબ્ધ બનાવી શકો છો અને વિવિધ ભાષાઓમાં સુવિધાવાળી બનાવી શકો છો.

**You said:**

i18n and i10n banne nu koi ek real time example aam gujarati ma pan english alphabet sathe aapo etle ke msg ma aapne je rite vat krta hoy e rite

**ChatGPT said:**

સમજી લેવા માટે, i18n (આંતરરાષ્ટ્રીયકરણ) અને l10n (સ્થાનિકીકરણ) ના સાચા-સમયના ઉદાહરણ સાથે તમને એ અંગ્રેજી અને ગુજરાતી ભાષામાં સરળ રીતે સમજાવું છું.

**i18n (Internationalization) અને l10n (Localization) નું એક રીયલ-ટાઇમ ઉદાહરણ:**

**Imagine kariye ek mobile application che, jemne duniya bhar ni loko use kare che.**

**1. i18n (Internationalization) -**

i18n ek prakriya chhe, je ma apne app ne aa rite banaviye chhe ke ene duniya bhar ni alag-alag languages, cultures ne samarthan karvi sake. Jyaare i18n thai chhe, tyare app na core logic ne aapde aavi rite design kariye chhe, je alag-alag countries ane cultures ma kaam karse.

**Example**:  
Maan lo apne ek weather app banavi rahi chhe je duniya bhar na users use kare chhe. Toh apne i18n na process ma apne ema date format, time format, currency, ane units ne flexible banaavva padse.

* **Date format**: India ma "dd/mm/yyyy" hoy, parantu USA ma "mm/dd/yyyy" hoy.
* **Time format**: India ma "24-hour" format use kariye chhe, parantu USA ma "12-hour" format use thay chhe.
* **Currency**: India ma INR (₹) use kare chhe, parantu USA ma USD ($) use thaye chhe.

**Code Example** (i18n Setup):

dart

Copy

import 'package:intl/intl.dart';

void main() {

DateTime currentDate = DateTime.now();

String formattedDate = DateFormat('dd/MM/yyyy').format(currentDate);

print("Formatted date: $formattedDate");

}

Aapne aa code thi apne date ne user ni region anusar change kari sakta chhiye. Aa apne i18n ne implement karvani ek misaal chhe.

**2. l10n (Localization) -**

l10n ee apne app ne specific language mate customize karvi chhe. Jyaare i18n setup thase, tyaare localization (l10n) karine apne actual messages ne alag-alag languages ma change kariye chhe.

**Example**:  
Apne weather app ma "Hello, welcome to Weather App!" message chhe. Toh apne aa message ne alag-alag languages ma convert karvu padse.

* **English**: "Hello, welcome to Weather App!"
* **Gujarati**: "હેલો, વેધર એપમાં આપનું સ્વાગત છે!"
* **Hindi**: "नमस्ते, वेदर ऐप में आपका स्वागत है!"

**Code Example** (l10n Setup):

dart

Copy

import 'package:flutter/material.dart';

import 'package:intl/intl.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

localizationsDelegates: [

GlobalMaterialLocalizations.delegate,

GlobalWidgetsLocalizations.delegate,

],

supportedLocales: [

Locale('en', 'US'), // English

Locale('gu', 'IN'), // Gujarati

Locale('hi', 'IN'), // Hindi

],

home: MyHomePage(),

);

}

}

class MyHomePage extends StatelessWidget {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Weather App'),

),

body: Center(

child: Text(

'Hello, welcome to Weather App!', // English message

),

),

);

}

}

Aapne aa code ma supportedLocales ni madad thi alag-alag languages (English, Gujarati, Hindi) specify kari che. Toh app aa language anusar message show kare che.

**Final Example - Combination of i18n and l10n:**

**Imagine kariye ek screen je ma weather info dikhai rahi chhe.**

* **India ma (Gujarati)**: "હવે તાપમાન: 32°C"
* **USA ma (English)**: "Current temperature: 90°F"

**i18n part**:

* Apne time format, currency, units, and locale set kariye chhe, je ena user na location ne anusar badlai sake.

**l10n part**:

* App message ne specific language ma translate karva mate, "હવે તાપમાન" ane "Current temperature" ne alag-alag languages ma customize kariye chhe.

**Conclusion:**

* **i18n** (Internationalization) na through, apne app ne globally usable banaavva mate flexible banaviye chhe.
* **l10n** (Localization) na through, apne app ne specific languages (like Gujarati, Hindi, English) ma adapt kariye chhe.

Aapda app ne i18n and l10n ma set karva thi, apne duniya na koi bhi user ne friendly anubhav aapi shakye chhiye.

Get smarter responses, upload files and images, and more.

Log in

Sign up for free

Top of Form



Attach

Search

Reason

Voice

Bottom of Form

ChatGPT can make mistakes. Check important info.

?