

ESAME DI INGEGNERIA DELLA CONOSCENZA



Documentazione progetto DIAGNOSI MMN

<https://github.com/RossellaSgaramella/Progetto-Icon.git>

SGARAMELLA ROSSELLA [717682]

r.sgaramella13@studenti.uniba.it

1.0 Introduzione

Diagnosi MMN è un sistema esperto in grado di riconoscere eventuali carenze di macro e micronutrienti nelle piante, simulando l'esperienza di un fisiologo vegetale o di un botanico esperto che guida e assiste l'utente.

Il suo compito è quindi quello di riconoscere una determinata carenza di un nutriente, arrivando ad una diagnosi e proponendo un'eventuale terapia per risolvere il problema, se trovato.

Viene quindi effettuata una prima analisi visiva dei problemi presenti sulle foglie. In seguito, se trovata una carenza di un qualche nutriente, viene chiesto di inserire i valori dell'elemento carente, recuperati dal terreno o dai tessuti fogliari.



2.0 PROGETTAZIONE

Per sviluppare questo progetto, ho ritenuto opportuno utilizzare il ragionamento con vincoli, in quanto una particolare carenza è definita da regole ben precise.

Ad ogni step, quindi, viene fatta una domanda all'utente, e sulla base della risposta di questo, il sistema accresce la sua conoscenza, andando quindi a fare riferimento a determinate regole piuttosto che ad altre.

Le regole per definire la carenza di un nutriente, sono state reperite da internet, ma anche da veri e propri esperti del settore. In questo modo, avendo due tipologie di fonti, avevo la certezza che le regole definite, fossero valide. Infatti, prima di procedere alla realizzazione del codice, ho creato una bozza di tutte le regole utili per effettuare una diagnosi.

Dopodiché, queste sono state revisionate dall'esperto e una volta avuta la sua approvazione, ho proceduto con la vera e propria implementazione del codice.

È stata anche creata una cartella (res), dove ho inserito diversi file di testo, uno per ogni nutriente per cui il sistema è in grado di diagnosticarne una carenza. In ognuno di questi file testuali, è riportata la funzione del nutriente nelle piante, e quali sono le conseguenze di ciò.

Il sistema, una volta diagnosticata la carenza di un nutriente, accede al file corrispondente e ne stampa il contenuto, in modo da poter anche informare l'utente. Fatto ciò, viene chiesto all'utente se desidera che gli venga proposta una possibile terapia. In caso di risposta positiva, questa viene stampata.

3.0 RAGIONAMENTO CON VINCOLI

Il sistema è basato su vincoli. Un vincolo è composto da una coppia <scope, relationship> che definisce quali sono le variabili sulle quali è imposto il vincolo e quale sia il vincolo imposto. È stato utilizzato un approccio di Domain Splitting, ovvero il problema è stato suddiviso considerando una variabile alla volta in esso, nel quale il valore della variabile è fissato. Abbiamo quindi diverse tipologie di regole:

- **Regole condivise**

Sono regole che valgono per tutti i tipi di foglia, dalle più giovani alle più vecchie. Queste regole vanno a verificare se sulle foglie c'è clorosi internervale, se ci sono aree clorotiche, o morte dei tessuti.

- **Regole che valgono per foglie giovani**

Sono regole apposite per le foglie giovani, in particolare verificano se le foglie presentano morte dei tessuti, un colore verde chiaro, forma ad uncino o crescita bloccata

- **Regole per le foglie vecchie**

Verificano se le foglie più vecchie hanno un colore verde scuro, se hanno degli spot rossastri o se sono arricciate

- **Regole per effetti localizzati**

Servono per gli effetti localizzati. Ad esempio, se ci sono foglie rosse o foglie che presentano bruciature o se c'è una separazione netta tra i tessuti vivi e quelli morti

- **Regole per l'analisi dai valori estratti dai tessuti fogliari**

Questo tipo di regole vengono utilizzate per raccogliere i valori del nutriente carente espressi in ppm recuperati dai tessuti fogliari o i valori dello stesso nutriente in mg/kg recuperati dal terreno

- **Regole per il calcolo % dei tessuti fogliari**

Queste regole servono per stabilire il tipo livello di carenza del nutriente (es. grave carenza, livello carente, livello normale, eccesso) basandosi sui valori recuperati dai tessuti fogliari

- **Regole per il calcolo % del terreno**

Sono regole utilizzate per stabilire il livello di carenza del nutriente nel terreno

3.1 REGOLE PER LA CARENZA DI AZOTO

Le piante con carenza di Azoto (elemento mobile) hanno una crescita rallentata. Le foglie più vecchie non possono produrre clorofilla e diventano gialle tra le nervature, mentre le nervature restano verdi. L'ingiallimento prosegue sull'intera foglia fino a farla cadere. Alle volte le foglie e i rami possono diventare rossicci, il che porta a *confondere la carenza di azoto con la carenza di fosforo*. L'azoto è un elemento molto mobile e viene consumato velocemente quindi va aggiunto regolarmente tramite concimi a lenta cessione.



- Foglie vecchie e ingiallimento tra le venature che può estendersi gradualmente
- Rami e venature delle foglie possono diventare rosso- porpora
- Le foglie giovani sviluppano una clorosi intervenale
- Caduta delle foglie

Cura: Fertilizzare con concimi a rapido assorbimento, come Cellmax Bio-Grow. I risultati si vedono in 4-5 giorni.

3.2 REGOLE PER LA CARENZA DI FOSFORO

Le piante con carenza di Fosforo hanno una crescita rallentata. Le foglie sono piccole, verde bluastro e spesso con macchie. Gli steli, le foglie e le nervature diventano rossastri a partire dai bordi e dal retro. L'arrossamento però non sempre è pronunciato. Le foglie vecchie diventano scure e arricciate e i bordi si incurvano. Se la carenza è grave le foglie sviluppano grandi macchie viola scuro che diventano poi bronzee, secche, raggrinzite e cadono. La carenza si aggrava in suoli argillosi o acidi. Quella più comune è quando il substrato ha un livello di pH superiore a 7. Con un livello di pH sotto a 5,8 e con eccessi di zinco e ferro, i fosfati risultano non assimilabili dalla pianta.



- Clorosi lungo le nervature fogliari (a partire dalle foglie giovani)
- Le foglie diventano bronzo-porpora, raggrinziscono e muoiono
- Foglie verde-bluastro scuro spesso con macchie scure
- Crescita stentata

Cura: Fertilizzare con concimi a rapido assorbimento come il P 20%. I risultati si vedono in 4-5 giorni.

3.3 REGOLE PER LA CARENZA DI POTASSIO

La carenza di potassio (elemento mobile) avviene occasionalmente sia nei substrati di coltivazione indoor che all'aperto in terreno, ma raramente nelle coltivazioni idroponiche. La pianta con carenza di potassio sviluppa sulle foglie dei bordi gialli e macchie clorotiche (simili alla carenza di calcio ma solo sul bordo fogliare), le foglie basse muoiono o diventano marrone chiaro. La pianta diventa suscettibile alle malattie. Il potassio è di solito presente nel suolo però, spesso, è bloccato dall'alta salinità del substrato stesso. Aumenta la temperatura interna delle foglie causando il degrado delle proteine delle cellule. L'eccessiva evaporazione dai margini fogliari causa bruciature rosse associate a steli dello stesso colore.



- Le foglie vecchie perdono lucentezza e sviluppano macchie rugginose
- I margini fogliari diventano secchi, marroni e si sollevano
- Fioritura ritardata e molto ridotta
- I rami e i gambi diventano molto sottili

Cura: Fertilizzare con concime a rapido assorbimento come il K 20%. I risultati si vedono in 2-4 giorni.

3.4 REGOLE PER LA CARENZA DI MAGNESIO

La carenza di Magnesio (elemento mobile) nelle piante è resa evidente solitamente nel mezzo della pianta o nelle foglie più basse, poi si diffonde verso le foglie più giovani. Le foglie diventano gialle o bianche con venature che rimangono di un verde scuro: questi gli indicatori più evidenti di una mancanza di magnesio. Le foglie più basse e quelle di mezzo sviluppano delle chiazze gialle tra le venature verde scuro. I margini fogliari si piegano prima che la foglia cada. La pianta può ingiallire nel giro di poche settimane e se è grave diventa completamente bianca. La carenza di magnesio non è comune nelle coltivazioni outdoor.



- Clorosi a chiazze (a partire dalle foglie vecchie)
- Perdita precoce delle foglie più vecchie
- Ripiegamento verso l'alto dei margini fogliari
- Chiazze giallognole o violacee sulla lamina fogliare

Cura: Fertilizzare con concime idrosolubile a rapido assorbimento come il MgO 8%. I risultati si vedono in 2-4 giorni.

3.5 REGOLE PER LA CARENZA DI CALCIO

La carenza di Calcio in una pianta può influenzare il suolo facendolo diventare troppo acido. È molto abbondante in natura e si trova in forma di calcare (CaCO_3). Le sue eventuali carenze si possono riscontrare dunque in suoli acidi. Sulle foglie si possono osservare deformazioni e clorosi internervali, oltre che disseccamenti degli apici vegetativi e dei margini fogliari.



- Clorosi delle foglie giovani, seguita da necrosi e caduta
- Colore delle foglie opaco, ingiallimento
- Foglie deformate
- Curvatura e appassimento dei lembi fogliari

Cura: Fertilizzare con concime a rapido assorbimento come il Mono Calcio 15%.

3.6 REGOLE PER LA CARENZA DI FERRO

La carenza di Ferro nelle piante incomincia dalle foglie nuove ed è tipicamente segnalata da clorosi internervali, che possono evolvere in scolorimenti diffusi delle foglie apicali. È comune quando il livello del pH è superiore a 6.5. I sintomi possono comparire durante i periodi di forte crescita o di stress. Le venature restano verdi mentre le zone interne ingialliscono. La clorosi diventa a mano a mano più acuta e, nei casi più gravi, le foglie cadono. Alcune piante, le acidofile, potrebbero avere dei problemi di carenza di ferro se piantate in terreni neutri o basici.



- Clorosi molto pronunciata delle foglie giovani a partire dalle nervature
- L'ingiallimento si diffonde anche sulle foglie più vecchie
- Si sviluppano delle necrosi delle foglie

Cura: Abbassare il pH del suolo ed evitare concimi che contengano Zinco o Manganese. Successivamente applicare chelati di Ferro come il FishPlant Iron.

3.7 REGOLE PER LA CARENZA DI MANGANESE

La carenza di Manganese (elemento non mobile) è rara e quasi sempre è associata alla carenza di ferro e zinco, si manifesta nelle zone intervenali che ingialliscono mentre le venature restano verdi. I sintomi poi si diffondono sulle foglie più vecchie. Potrebbero formarsi delle zone necrotiche che si sviluppano sulle foglie molto colpite che poi cadono. La pianta appare appassita.



1. Clorosi intervenali sulle foglie nuove
2. Sulle foglie molto colpite si sviluppano zone necrotiche
3. La crescita è rallentata e gli apici sono raggrinziti

Cura: Integrare nella fertirrigazione con concimi ricchi di microelementi, come il Revive e Ionic Cal Mag.

4.0 REALIZZAZIONE

Dato l'elevato numero di domande che possono essere poste all'utente, è stato definito un metodo per validare le risposte. Questo riceve come parametri la domanda e le risposte valide per questa domanda. Facendo così, si garantisce che l'utente inserisca sempre risposte valide. Allo stesso modo, ho implementato un metodo che ha lo stesso compito del precedente, con la differenza che deve validare i valori numerici inseriti.

Vengono inoltre definite le varie terapie, una per ogni nutriente carente.

Passando al vero e proprio funzionamento, viene mostrato un messaggio di benvenuto, la prima domanda che viene poi posta all'utente riguarda la tipologia di foglie colpite (giovani, vecchie o entrambi). Partendo da questo presupposto, il sistema pone altre domande, in modo da accrescere la propria conoscenza. In base alle risposte ricevute, il sistema si riferisce a determinate regole. Dopo aver posto diverse domande sull'aspetto visivo della pianta, il sistema effettua una diagnosi, per rilevare il nutriente di cui la pianta soffre.

Fatto ciò, viene chiesto all'utente se ha a disposizione i valori del nutriente, recuperati dal tessuto fogliare e/o dal terreno. Questi dati vengono utilizzati per calcolare il livello di carenza del nutriente (grave, basso, normale, eccesso) nel tessuto fogliare e/o nel terreno.

Viene infine proposta una possibile cura, con anche delle raccomandazioni sul prodotto da utilizzare.

Può anche succedere che il sistema non rilevi alcuna diagnosi.

4.1 LIBRERIA EXPERTA

La libreria utilizzata per costruire l'applicazione basata su regole, è Experta. In Experta una regola è un richiamabile, denominata con Rule.

Le regole hanno due componenti, LHS (lato sinistro) e RHS (lato destro):

1. L'LHS descrive (utilizzando modelli) le condizioni in cui la regola * dovrebbe essere eseguita (o licenziata).
2. Il RHS è l'insieme di azioni da eseguire quando la regola viene attivata.

La libreria si basa sui Facts, i quali sono l'unità base di informazione per Experta. Affinché un Facts corrisponda a un Modello, tutte le restrizioni del modello devono essere Vere quando il Facts viene valutato rispetto ad esso.

Un fact, infatti, definisce una condizione:

```
Fact(clorosi_internervale="si"),
```

È possibile utilizzare operatori logici per esprimere condizioni LHS complesse.


```

@Rule(
    AND(
        Diagnosi(nome=MATCH.nome),
        OR(
            TEST(lambda nome: nome == "Carenza Boro"),
            TEST(lambda nome: nome == "Carenza Cloro"),
            TEST(lambda nome: nome == "Carenza Rame"),
            TEST(lambda nome: nome == "Carenza Ferro"),
            TEST(lambda nome: nome == "Carenza Manganese"),
            TEST(lambda nome: nome == "Carenza Molibdeno"),
            TEST(lambda nome: nome == "Carenza Zinco"),
        ),
        NOT(Fact(campione_tessuti=W())),
    ),
    salience=3
)
def regola_27(self, nome):
    self.declare(Fact(campione_tessuti=si_o_no(
        "Hai a disposizione i valori di {} in ppm recuperati da tessuti fogliari".format(get_nome_elemento(nom

```

Per definire una regola, dobbiamo definirne tutti i Facts:

```

@Rule(OR(
    AND(
        Fact(foglie_colpite="giovani"),
        Fact(morte_tessuti="no"),
        Fact(clorosi_internervale="si"),
        Fact(venature_distinte="no"),

```

In base allo stato del mondo rilevato, viene posta una determinata domanda all'utente, per ricavare nuove informazioni. Si utilizza, infatti, la clausola NOT, quando si vuole chiedere all'utente qualcosa che non si conosce:

```

@Rule(OR(
    AND(
        Fact(foglie_colpite="giovani"),
        Fact(morte_tessuti="no"),
        Fact(clorosi_internervale="si"),
        Fact(venature_distinte="no"),
        NOT(Fact(spot=W()))

```

E viene fatta la domanda:

```
def regola_4(self):
    self.declare(Fact(spot=si_o_no(
        "Sulla foglia sono presenti aree necrotiche (spot)")))
```

Una volta che il sistema dispone di tutte le informazioni necessarie, la diagnosi viene effettuata in base alle regole stabilite per ogni nutriente:

```
@Rule(
    AND(
        Fact(foglie_colpite="giovani"),
        Fact(morte_tessuti="si"),
        Fact(colore_verde_pallido="si"),
        Fact(annicciate="si"),
        Fact(cura_boro=MATCH.cura_boro)
    )
)
def diagnosi_carenza_boro(self, cura_boro):
    self.declare(Diagnosi(
        nome="Carenza Boro",
        descrizione=leggi_da_file("boro")
    ))
    print("E' stata riscontrata una carenza di boro.")
    self.declare(Terapia(nome="Cura carenza Boro", descrizione=cura_boro))
```

Regole per la carenza del nutriente

Crea oggetto di classe Diagnosi, con nome e descrizione (letta da file)

Richiama la terapia corrispondente al nutriente

Se l'utente inserisce i valori del nutriente rilevati dal tessuto fogliare o dal terreno, per rilevare il livello di carenza vengono utilizzate altre regole:

```

@Rule(
    Fact(campione_terreno="si"),
    Fact(percentuale_terreno_boro=MATCH.percentuale_terreno_boro),
    TEST(Lambda percentuale_terreno_boro: float(percentuale_terreno_boro) >= 0.5),
    NOT(Fact(livello_terreno=W()))),
    salience=2
)
def terreno_carenza_boro_4(self):
    self.declare(Fact(livello_terreno="eccesso"))

```

La differenza tra Fatti e Modelli è piccola. In effetti, i pattern sono solo fatti contenenti elementi condizionali del pattern invece di dati regolari. Sono usati solo nella parte sinistra di una regola.

Se non fornisci il contenuto di un modello come PCE, Experta racchiuderà automaticamente il valore in un LiteralPCE per te.

La maggior parte delle volte i sistemi esperti richiedono la presenza di una serie di fatti affinché il sistema funzioni. Questo è lo scopo del decoratore DefFacts. Tutti i DefFacts all'interno di un KnowledgeEngine verranno chiamati ogni volta che viene chiamato il metodo reset:

```

class ExpertSystem(KnowledgeEngine):
    @DefFacts()
    def _initial_action(self):
        yield Fact(inizio='si')
        yield Fact(cura_fosforo="Fertilizzare con concimi a rapido assorbimento come il P 20%. I risultati si vedono dopo 10 giorni.")
        yield Fact(cura_azoto="Fertilizzare con concimi a rapido assorbimento, come Cellmax Bio-Grow. I risultati si vedono dopo 10 giorni.")
        yield Fact(cura_potassio="Fertilizzare con concime a rapido assorbimento come il K 20%. I risultati si vedono dopo 10 giorni.")
        yield Fact(cura_magnesio="Fertilizzare con concime idrosolubile a rapido assorbimento come il MgO 8%. I risultati si vedono dopo 10 giorni.")
        yield Fact(cura_calcio="Fertilizzare con concime a rapido assorbimento come il Mono Calcio 15%. Altri fertilizzanti a base di calcio sono disponibili.")
        yield Fact(cura_ferro="Abbassare il pH del suolo ed evitare concimi che contengano Zinco o Manganese. Successivamente, integrare con ferro.")
        yield Fact(cura_manganese="Integrare nella fertirrigazione con concimi ricchi di microelementi, come il R 1000.")
        yield Fact(cura_boro="Fertilizzare con concime idrosolubile a rapido assorbimento come il BORFAST. I risultati si vedono dopo 10 giorni.")
        yield Fact(cura_zolfo="In caso di fenomeni di carenza e' possibile aggiungere zolfo inorganico con l'aiuto di fertilizzanti a base di zolfo.")

```

KNOWLEDGE ENGINE

Il primo passo è creare una sottoclasse (KnowledgeEngine) e utilizzare Rule per decorare i suoi metodi.

Successivamente, si può creare un'istanza, popolarla con fatti e infine eseguirla.

```
engine = ExpertSystem()
engine.reset()
engine.run()
```

PROCEDURA DI ESECUZIONE DI ENGINE

Questo è il solito processo per eseguire un KnowledgeEngine.

1. La classe deve essere istanziata, ovviamente.
2. Il metodo di ripristino deve essere chiamato:
 - Questo dichiara il fatto speciale InitialFact. Necessario per il corretto funzionamento di alcune regole.
 - Dichiara tutti i fatti prodotti dai metodi decorati con @DefFacts.
3. Il metodo run deve essere chiamato. Questo avvia il ciclo di esecuzione.

In uno stile di programmazione convenzionale, il punto iniziale, il punto finale e la sequenza delle operazioni sono definiti esplicitamente dal programmatore. Con Experta, il flusso del programma non deve essere definito in modo così esplicito. La conoscenza (Regole) e i dati (Fatti) sono separati e il KnowledgeEngine viene utilizzato per applicare la conoscenza ai dati.

Il ciclo di esecuzione base è il seguente:

1. Se il limite di attivazione della regola è stato raggiunto, l'esecuzione viene interrotta.
2. La prima regola è selezionata per l'esecuzione. Se non ci sono regole, l'esecuzione viene interrotta.
3. Vengono eseguite le azioni RHS della regola selezionata (viene richiamato il metodo). Di conseguenza, le regole possono essere attivate o disattivate.

ESEMPI DI REGOLE UTILIZZATE ALL'INTERNO DEL CODICE:

REGOLE CONDIVISE

```

@Rule(Fact(inizio="si"))
def regola_1(self):
    print("DIAGNOSI MMN")
    print("Un sistema esperto per riconoscere le carenze nutrizionali nelle coltivazioni, basandosi in una fase  

    "raccolti da tessuto fogliare o dal terreno. ")
    self.declare(Fact(benvenuto="si"))

```

REGOLE FOGLIE GIOVANI

```

@Rule(AND(
    Fact(foglie_colpita="giovani"),
    Fact(morte_tessuti="si"),
    Fact(colore_verde_pallido="si"),
    NOT(Fact(arricciate=W()))
))
def regola_7(self):
    self.declare(Fact(arricciate=si_o_no(
        "Le foglie apicali sono arricciate")))

```

REGOLE FOGLIE VECCHIE O BASALI

```

@Rule(AND(
    Fact(foglie_colpita="vecchie"),
    Fact(colore_verde_scuio="si"),
    NOT(Fact(spot_rossastri=W()))
))
def regola_18(self):
    self.declare(Fact(spot_rossastri=si_o_no(
        "Sono presenti degli spot rossastri sulle foglie")))

```

REGOLE EFFETTI LOCALIZZATI

```

@Rule(AND(
    Fact(foglie_colpita="entrambe"),
    Fact(clorosi_internervale="si"),
    NOT(Fact(foglie_rosse=W()))
))
def regola_22(self):
    self.declare(Fact(foglie_rosse=si_o_no(
        "Sono presenti foglie rosse")))

```

REGOLE ANALISI VALORI ESTRATTI DA TESSUTI FOGLIARI

```

@Rule(
    AND(
        Diagnosi(nome=MATCH.nome),
        OR(
            TEST(lambda nome: nome == "Carenza Boro"),
            TEST(lambda nome: nome == "Carenza Cloro"),
            TEST(lambda nome: nome == "Carenza Rame"),
            TEST(lambda nome: nome == "Carenza Ferro"),
            TEST(lambda nome: nome == "Carenza Manganese"),
            TEST(lambda nome: nome == "Carenza Molibdeno"),
            TEST(lambda nome: nome == "Carenza Zinco"),
        ),
        NOT(Fact(campione_tessuti=W())),
    ),
    salience=3
)

def regola_27(self, nome):
    self.declare(Fact(campione_tessuti=si_o_no(
        "Hai a disposizione i valori di {} in ppm recuperati da tessuti fogliari".format(get_nome_elemento(nom

```

REGOLE CALCOLO % TESSUTI FOGLIARI

```

@Rule(
    Fact(campione_tessuti="si"),
    Fact(Fact(percentuale_tessuti_boro=MATCH.percentuale_tessuti_boro)),
    TEST(lambda percentuale_tessuti_boro: float(percentuale_tessuti_boro) < 5),
    NOT(Fact(livello_tessuti_fogliari=W())),
    salience=3
)

def tessuto_carenza_boro_0(self):
    self.declare(Fact(livello_tessuti_fogliari="grave carenza"))

```

REGOLE CALCOLO % TERRENO

```

@Rule(
    Fact(campione_terreno="si"),
    Fact(percentuale_terreno_fosforo=MATCH.percentuale_terreno_fosforo),
    TEST(lambda percentuale_terreno_fosforo: float(percentuale_terreno_fosforo) < 7),
    NOT(Fact(livello_terreno=W())),
    salience=2
)

def terreno_carenza_fosforo_0(self):
    self.declare(Fact(livello_terreno="grave carenza"))

```

5.0 CONSIDERAZIONI

Il sistema è in grado di rilevare una serie di carenze di diversi nutrienti.

Sono stati effettuati vari test sulle tipologie di carenze previste. In tutti i casi il sistema è stato in grado di riconoscere correttamente il nutriente carente. A patto che questo sia “supportato”.

Ho sempre riscontrato casi di true positive e true negative. I risultati quindi sono stati sempre veritieri ed accurati. Dato il limitato numero di carenze rilevabili, può capitare che il sistema non sia in grado di rilevare una determinata carenza in base ai valori inseriti da input.

Ho infatti deciso di non aggiungere ulteriori vincoli per rilevare altri nutrienti, perché si ritiene che questi siano i più importanti, oltre che i più comuni per i quali si rilevano effetti visibili nelle piante.

Tuttavia, il sistema è stato costruito in modo tale da poter facilmente implementare il riconoscimento di nuove carenze. Infatti, basta definire nuove regole apposite.

BIBLIOGRAFIA

1. https://www.idroponica.it/carenze-nutrizionali-piante-indoor-outdoor_28-175.html