

#Phonetics

1. 말할 때 허파에서 바람이 나온다. 바람이 나오는 것은 공기의 흐름인데, 폐가 고기압을 만들어 공기를 보내 성대에서 소리가 만들어진다. 성대가 떨면서 바람이 멈췄다가 나갔다가 하는데, 이는 파도와 비슷하다. 목에서 인위적으로 공기압을 만들어내서, 성대가 기문을 열고 막는다. 발성하는 순간, whisper하는 순간 기문이 완전히 열린다.

기문이 부드러운 근육이라 막아도 움직여서 진동이 느껴지는 것, 남자는 1초에 100번, 여자는 200번. 소리를 내라고 하면 동일한 톤의 소리를 내는데, 자연스레 무리로 동작하는 것을 sink라고 한다.

소리를 바꿀 수 있는 것은 성대가 아닌 입 모양. 성대는 특정 소리의 높이에 맞춰 있고, 입 모양은 혀의 위치와 턱도 함께 말한다. 그런데 우리는 턱의 높이를 일정하게 하고, 입 모양만 다르게 할 수 있다. 턱의 높낮이가 절대 변하지 않는 상황에서, 입 속에 있는 모양을 자유롭게 바꿔서 소리를 낼 수 있다. 턱의 높낮이가 소리를 바꾸는 주된 요인이 아니라는 것.

(한국어는 음절, 영어는 stress를 기준으로 시간이 반복된다. 한국어는 턱을 위주로 발음해 혀가 그냥 턱 위에 얹혀서 발음되는 것. Stress를 위주로 하는 영어는 턱보다 혀를 더 자유롭게 사용한다.)

➔ Articulation은 speech를 만드는 가장 첫번째 메커니즘.

공기압이 성대에서 1초에 100번 파도처럼 움직일 때 입 모양을 바꿔가면서 입 밖으로 소리를 내는 것. 사람에게 수반되는, 사람이 소리를 만들어내는 것에 대한 것.

2. 공기를 타고 가는 과정을 acoustic이라고 한다. 공기가 어떻게 움직이는, 공기와 소리에 대해 사람이 계획되지 않은 완전 물리화적인 것(acoustic physis). 사람의 입을 떠나서 공기를 타고 가는 원리.
3. Auditory는 귓바퀴를 통해, 입 밖으로 나온 소리를 듣는 것. 마이크로한 공기의 움직임을 잘 느끼게 하기 위해 소리를 빨아주는 귓바퀴. 공기가 귀로 들어가고 말을 할 때는 성대를 움직이고, 귀로 들을 때는 귓바퀴를 이용해 고막이 움직인다. (고막이 움직이는 것도 물리화적인 메커니즘) 파도가 고막을 120번 치는 것이 그대로 청각세포에 전달, 뇌, 신경에 영향을 주는 메커니즘. 사람이 수반된, 어떻게 듣는가에 대한 메커니즘.

#guitar plucking

기타가 어떤 음을 치든지 pure tone(sine wave 처럼).. 사람 목소리의 source 는 기타와 비슷하다. Pure tone 과 guitar tone 을 비교해보면, guitar tone 은 complex tone 이다. perceive 하는 음의 높이는 동일하다 – pure tone 이든 guitar tone 이든.

#synthesizing source [1 번째]

반복되는 패턴의 주기가 우리가 만들었던 총 10개의 sine wave 중에서 f_0 의 sine wave 와 일치한다. 이 소리를 듣고 여러분들이 인지심리학적으로 이 소리가 어떤 높이하고 일치할까요? 분명히 1000 헤르츠까지 포함됐는데, 이게 정확하게 인지청각적으로 100Hz 와 일치하다, 높이가 일치한다고 인식하게 된다. 1000Hz 높이는 사실 인지적으로 들리지 않는다. 그래서 f_0 이라고 하는 것이고, 우리의 pitch 라고 하는 것.

만약 무한대로 하게 되면, 어떻게 될까? 원래는 부드럽게 봤던 f_0 의 sine wave 가 점점 한 쪽으로 몰려 있는 게 보일 것. 한 쪽만 peak 있고, 다른 쪽만 peak 있고 그러던 것이, 무한대로 가면 peak 1 개 00000, peak 1 개 00000, 이런 식으로 된다. 이것을 우리가 curse 라고 부르고, 기차처럼 나온다고 해서 curse train 이라고 한다. Voice source 우리의 성대에서 만들어진, train 처럼 나온다.

#vowel space & formants [맨 마지막 슬라이드]

서로 다른 모음이 서로 다른 입모양을 가지고 있죠, 각각의 f_1 과 f_2 를 그래프로 나타낼 수 있는데, 어떻게 나타내냐면 아까 formants 그림에서 (각각의 값들이 있는데, 이것들을 f_2 를 x 축, f_1 을 y 축으로 해서 점을 찍어볼 수 있다. 어차피 두 값을, x,y 해서 graph 를 만든다.) 뭐랑 일치하냐면 i 모음을 발화해보세요, 입의 흐름과 동일하다. 위에서 아래로! x 축이 앞쪽이라고 생각해보고, (front-back), y 축은 높이라고 보면 된다.

F_1 은 모음(혀)의 높낮이, height? Height? F_2 는 front-back 을 결정한다.

New>sound>create sound from vowel editor

F_1 - f_2 plot 이 나왔고 각각의 모음들이 찍혀있다. 이것은 f_1 - f_2 plot 이다. F_1 은 y 축, f_2 는 x 축.

한국어 아와 영어 아가 어떻게 다를까?

2 차원적으로 하이트와 백니스를 이야기했는데 영어가 더 back, low 하다. Spa~ 훨씬 더 입을 크게 벌리고 훨씬 더 우리가 갖고 있는 입의 구조를 더 크게 이용하는 것.

한국어는 조금 더 좁게 이용을 하는 것.

#코딩

영어 뿐만 아니라 일반적인 언어, 소리가 어떻게 표현되고 어떤 원리로 구성되는가. 아무리 복잡한 소리라도 여러 sine wave 들의 합으로 이루어지는 것을 praat 으로 이용해봤고, frequency 에서 formant(f1,f2)가 어떻게 모음을 구별하는가. 사실은 이론을 배웠고 누가 짜놓은 프로그램을 쓴 것이죠. 이론적으로 우리가 실제로 어떤 사람들이 코딩을 해서 만든 software program 을 이용한 것뿐. 목적이나 방향이 코딩을 배운다는 것이 어떤 것인지를 이해하면 된다. 조금 더 응용해서 praat 같은 것을 만들어내는 것.

코딩만 주구장창 이야기하는 것은 이 수업의 의미가 없기에, 코딩을 아주 간단하게 앞에서 훑고 진도를 뺄 것이다.

코딩은 기본적으로 자동화라고 생각하면 된다. 여러분들이 이론으로 엄청나게 배웠고 praat 을 할 때 자동화가 되어서 프로그램 내부적으로 무언가 계산이 된 것이다. 자동화라는 것은 왜 자동화를 할까요? 라고 생각을 해볼 때, 똑같은 게 반복이 안되면 자동화할 필요가 없는데, 자동화를 한다는 이유는 똑같은 게 계속 반복되니까 자동화를 시킨 것, 마찬가지로 코딩도 자동화를 하는 것이다. 직접 다 할 수 있지만, sine wave 들이 숫자들이라서 그것을 더하면 complex tone 이 되는 것인데, 이게 한 번만 하는 게 아니라 여러 번 계속 하는 거니까 그것을 코딩으로 만든 것. 계속 사용할 거라서 반복되는 것을 자동화해준다는 측면에서 우리는 코딩을 한다고 생각하면 된다. 여러분 폰의 모든 것, 컴퓨터에서 쓰는 모든 것들, 프로그램들 하나하나가 다 코딩으로 이루어지지 않을 수가 없다. 누구한테 자동화를 시키냐면 컴퓨터에 자동화를 시키는 것. 컴퓨터에 이렇게 해라 저렇게 해라 자동화를 시키는 것을 코딩이라고 생각하면 된다. 컴퓨터, 태블릿 pc, 스마트폰도 마찬가지다. 컴퓨터에게 시키는 언어가 코딩, program language 라고 생각하면 된다.

모든 언어는 공통적으로 단어가 있다, 그것을 combine 하는 것으로 이루어져 있다. 컴퓨터 언어와 사람의 언어가 같은 특징을 가지고 있으니까 말하는 것이다. 단어는 특징이 무엇일까? 단어는 그 속에 무엇을 포함하고 있죠? 의미이죠. 의미는 정보가 들어간 것. 정보를 담는 어떤 그릇 같은 것이라 생각하면 된다. 단어라는 그릇이 하나 크게 있다. 그릇에 이런 정보를 담으면 이런 단어가 되는 것. 그릇이 딱 하나 있고, 그 그릇에 어떤 정보가 담기느냐 따라 단어가 달라지는 것.

단어를 먼저 선택하고 어떻게 combine(순서도 중요함, 단순히 결합하는 것이 아님)하는지. 문법 혹은 combination 으로 문장을 만들어 의미를 표현한다. 동일하게 컴퓨터 언어도 그렇다. 컴퓨터 언어의 단어에 해당하는 부분이 변수이다. 영어로는 variable 이라고 한다. 아까 단어의 역할이 정보를 담는 그릇이라고 했지, 정보를 담는 그릇이 변수로서 필요한 것. 의사소통할 때 문법이 필요하다. 기계와의 문법은 생각보다 복잡하지 않다. 여러분들이 문법에 대해 아셔야 할 게 세가지 정도 있는데, 첫번째 변수라고 이야기했던 그릇에다 정보를 넣는 것, 처음에는 그냥 빈 그릇을 가지고 있고 거기에다 정보를 assign 하는 것. 변수에다 정보를 assign 하는 것. 두번째는 자동화, 기계화라고 생각할 때 그냥 직관적으로 떠오르는 게 뭐냐면, ~할 때는 이렇게 하라는

조건이 붙는게 당연히 필요하다. 그러한 conditioning 에 대한 문법이 필요하다. if 문법을 씁니다 실제로. 세 번째는 자동화의 가장 중요한 것 중에 하나가 여러 번 반복하는 것. 이것을 for 문법을 써서 여러 번 반복하는 것을 가능하게 한다.

첫 번째, 정보 assign, 두번째 if 문법, 세번째 for loop 를 이용해 반복시키는 것. 이 세가지가 프로그램의 공통적 언어라고 생각하면 된다. 네번째가 사실 가장 중요한데, 지금까지 이야기했던 건 개별적인 문법인데, 지금 그걸 우리가 어떤, 결론만 먼저 말을 하면 함수를 배워야 한다. 문법에서 제일 중요한 게 함수를 배우는 건데, 함수란 중간에 무엇이 들어가는지는 모르겠지만 어떤 값을 입력하면 특정 값이 출력되는 것이라고 생각하면 된다. praat 에서 마우스를 움직였을 때 소리가 났지, 마우스의 지그재그 클릭이 입력이었고, 출력은 play 된 소리. 내부적으로 함수가 적용이 된 것이겠지. Variable assignment, if, for loop. 한꺼번에 입력해서 package 해서 출력하는 것을 함수라고 한다.

예를 들어서 어떤 게 함수가 될 것인지, 내가 무엇을 가지고 함수를 만들 수 있을지 생각을 해보세요. 함수라는 정의가 내부적으로 뭐가 들어가는지는 모르겠지만 분명히 들어가서 input 되서 값이 출력되는 것이 함수라고 했죠. 인간 자체가 함수라고 한다면, 그것에 대해 동의할 수 있을까요? 아침에 먹고 밤에 가서 화장실을 가죠, 전형적인 함수의 예죠. 입력이 들어가서 바뀌고 출력이 되는 것. praat 에서 쓴 것도 함수고, 자동차 생각해보면 입력 들어가고, 자동차라는 함수가 있을 건데 입력을 바꿔서 움직임으로 출력되는 것. 예를 들어 두개의 자연수를 주면 그 시작부터 끝까지 자연수만큼 증가를 해서 총 합을 하도록 해라, 7 과 10 을 주면 답이 뭐가 될까? 그리고 자연수를 7 에서 10 까지 자연수를 합하면 34 나오잖아. 함수도 마찬가지로 for loop 로 입력만 바꿔서 계속 반복되게 한다. 똑 같은 식만 넣어도 나올 것이 예상되는데, 함수라고 하죠. 함수 속에 들어가는 것들이 별 게 아니고 복잡한 것 같아도 아까 이야기했던 variable assignment, if condition, for loop 이런 것들만 있고, 함수 속에도 함수가 들어갈 수 있겠지. 재사용도 가능하고 반복적으로 사용이 되는게 코딩할 때 해야 하는 일이고, 그래서 코딩을 배워야 한다.